Journal of Bionic Engineering (2023) 20:1153–1174
https://doi.org/10.1007/s42235-022-00298-7

RESEARCH ARTICLE

Crisscross Harris Hawks Optimizer for Global Tasks and Feature Selection

Xin Wang[1] · Xiaogang Dong[1] · Yanan Zhang[2,3] · Huiling Chen[4]

Received: 3 February 2022 / Revised: 19 October 2022 / Accepted: 20 October 2022 / Published online: 30 November 2022

Abstract

Harris Hawks Optimizer (HHO) is a recent well-established optimizer based on the hunting characteristics of Harris hawks, which shows excellent efficiency in solving a variety of optimization issues. However, it undergoes weak global search capability because of the levy distribution in its optimization process. In this paper, a variant of HHO is proposed using Crisscross Optimization Algorithm (CSO) to compensate for the shortcomings of original HHO. The novel developed optimizer called Crisscross Harris Hawks Optimizer (CCHHO), which can effectively achieve high-quality solutions with accelerated convergence on a variety of optimization tasks. In the proposed algorithm, the vertical crossover strategy of CSO is used for adjusting the exploitative ability adaptively to alleviate the local optimum; the horizontal crossover strategy of CSO is considered as an operator for boosting explorative trend; and the competitive operator is adopted to accelerate the convergence rate. The effectiveness of the proposed optimizer is evaluated using 4 kinds of benchmark functions, 3 constrained engineering optimization issues and feature selection problems on 13 datasets from the UCI repository. Comparing with nine conventional intelligence algorithms and 9 state-of-the-art algorithms, the statistical results reveal that the proposed CCHHO is significantly more effective than HHO, CSO, CCNMHHO and other competitors, and its advantage is not influenced by the increase of problems' dimensions. Additionally, experimental results also illustrate that the proposed CCHHO outperforms some existing optimizers in working out engineering design optimization; for feature selection problems, it is superior to other feature selection methods including CCNMHHO in terms of fitness, error rate and length of selected features.

Keywords Harris hawks optimization · Bioinspired algorithm · Global optimization · Engineering optimization · Feature selection

1 Introduction

Optimization is the procedure of determining the optimal solution for a specific issue with a reasonable computational expense. It has received increasing attentions in a variety of scientific fields of research and engineering and is essential to propose splendid optimization algorithms [1–4]. Traditional deterministic optimization approaches cannot mostly feasible in each real-world problem [4]. As becoming more and more sophisticated, the various optimization issues are extensively solved using the promising stochastic bioinspired algorithms [5–9]. This kind of algorithm is unsensitive to the searching magnitude and has strong optimization capability and flexibility of finding high-quality solution efficiently through a brief process, even for high-dimensional difficult problems [4, 10–18]. A great number of well-regarded bioinspired optimizers, such as Differential Evolution (DE) [19], Particle Swarm Optimization (PSO)

✉ Xiaogang Dong
dongxiaogang@ccut.edu.cn

✉ Huiling Chen
chenhuiling.jlu@gmail.com

Xin Wang
wangx@ccut.edu.cn

Yanan Zhang
yananily@stu.xjtu.edu.cn

[1] School of Mathematics and Statistics, Changchun University of Technology, Changchun, Jilin 130012, China

[2] School of Management, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China

[3] Information Construction Office, Changchun University of Technology, Changchun, Jilin 130012, China

[4] College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China

[8], Sine Cosine Algorithm (SCA) [20], Grey Wolf Optimizer (GWO) [21], Moth-Flame Optimization (MFO) [22], Whale Optimization Algorithm (WOA) [23], Bat Algorithm (BA) [24], Gravitational Search Algorithm (GSA)[25], Harris Hawks Optimizer (HHO) [26], Slime Mould Algorithm (SMA) [27], Beluga Whale Optimization (BWO) [28], Dandelion Optimizer (DO) [29], Marine Predators Algorithm (MPA) [30], and their variants have been presented to handle a various optimization problems.

HHO is a newly stochastic bioinspired optimization algorithm with promising performance of dealing with continuous problems. It can provide easy implement and outstanding exploitative capability of local search. The algorithm is developed according to the inspiration of cooperative preying behavior of Harris' hawks and skilled in converging fast and making a suitable balance between intensification and diversification. The hawks' dynamic chasing patterns, which are resulted from the natural surrounding and time-vary escaping ways of prey, are composed of six phases, two in exploration phase and the remainder in exploitation phase. With the help of the stochastic operations of six phases, HHO can find excellent solutions compared to other well-regarded algorithms such as Genetic Algorithm (GA) [31], Biogeography-Based Optimization (BBO) [31], DE [31], PSO [31], Cuckoo Search algorithm (CS) [32], TLBO [33], BA/BAT [24], FPA [34], FA [35], GWO [21], and MFO [22]. Additionally, it can be superior to other optimizers in constrained engineering optimization tasks. HHO has already attracted many attentions in a variety of application fields soon after its occurrence [4, 36–49]. These superiority and meaningful applications reveal that HHO is a powerful enough optimizer. Notwithstanding, as a stochastic bioinspired algorithm, HHO also may undergo the shortcomings of trapping into local optima and degrading convergence rate when tackling some complex tasks. On one hand, it undergoes weak global search capability because of the levy distribution in its optimization process, which makes it difficult to run away from the local optima. On the other hand, the strong randomness in exploration phase may result in skipping the search spots at the edge of search space. Third, the "No-Free-Lunch" (NFL) theorem indicated that no optimizer can be the best general approach for all kinds of problems. With the motivation of these considerations, this study employs the Crisscross Optimization Algorithm (CSO) strategy as an operator to improve HHO.

CSO [50] is a well-established stochastic algorithm inspired by crossover behaviors based on the principle of the gold mean in Confucian doctrine. As proposed in its original literature, the simple CSO offers fast convergence rate and high-quality solution with preserving the population diversity when solving not only continuous but also complex optimization problems. A dual crossover search behavior in CSO is employed to establish an interacting chain of horizontal crossover operator and vertical crossover operator. As a global optimization operator, horizontal crossover conducts a crossing over arithmetically on two distinct search agents across all their dimensions, which are, respectively, chosen from the subpopulations without repetition. This operation ensures a larger probability of searching in each subspace and diminishes effectively the unreachable search spots. Vertical crossover is to exchange the dimensional information of a single search agent, which can facilitate escaping from the stagnancy in local optima without destroying other dimensions that may be the global optimum. Considering the advantages of CSO, some papers tried to apply this algorithm on handling complex problems [50–57]. In addition, the excellent search capability of the two crossover operators can exactly make up for the deficiencies of HHO mentioned above. Therefore, this study proposes an improved HHO using these two powerful crossover mechanisms in CSO. The proposed optimizer is called CrissCross Harris Hawks Optimizer (CCHHO).

CCHHO contains three phases. After the first initialization phase, the vertical crossover as a regulation mechanism in the second phase interacts with the exploitative behavior of HHO to modify partial search agents for adjusting the exploitative ability adaptively. With the help of dimensional mutation of vertical crossover, the excessive local search capability of the algorithm can be alleviated. In the third phase, the horizontal crossover is considered as an operator to boost explorative trend by diminishing the unreachable search spot. Moreover, to accelerate the convergence speed, the competitive operator conducts greedy selection after performing each crossover. The introduction of three operators in CSO into HHO can enhance the equilibrium between exploratory and exploitative tendencies in case of an accelerated convergence speed and unchanged computational complexity. Furthermore, to access the effectiveness of CCHHO, we take a series of comprehensive experiments from three kinds of issues using 4 categories of benchmark functions, 3 well-regarded engineering optimization tasks and feature selection problems from 13 UCI datasets. On the global optimization tasks of benchmark functions, to show the superiority of the proposed optimizer more rigorously, the performance of the improved optimizer on the global optimization problems of benchmark functions is, respectively, compared with that of nine well-known bioinspired optimizers and that of nine advanced algorithms. 9 well-known bioinspired optimizers contains DE [19], PSO[8], SCA [20], GWO [21], MFO [22], WOA [23], B) [24], GSA [25] and CSO[50], which is used as a strategy in the developed algorithm. 9 advanced algorithms consist of LSHADE_cnEpSin [58], SaDE [59], LSHADE[60], CGPSO [61], CLPSO [62], ALCPSO [63], ACWOA [64], IWOA [65], and CCNMHHO [49], which is the variant of HHO based on CSO and Nelder-Mead simplex. And then the scalability evaluation and sensitivity analysis of CCHHO are conducted. Besides that, the CCHHO optimizer is compared with some other previous engineering optimizers when dealing with engineering optimization problems. On feature selection, the modified HHO is discretized to construct a feature selection method, which is evaluated by comparing with other feature selection techniques including CCNMHHO. The experimental results reveal that the proposed CCHHO is significantly more effective than HHO, CSO, CCNMHHO and other competitors on diverse categories of functions, and its advantage is not influenced by the increase of problems' dimensions. Additionally, the results also illustrate that the proposed CCHHO outperforms some existing optimizers in working out engineering design optimization; for feature selection problems, it is superior to other feature selection methods including CCNMHHO in terms of fitness, error rate and length of selected features. Notably, the proposed CCHHO in this study and CCNMHHO both adopt the horizontal crossover and vertical crossover of CSO, while CCNMHHO is modified one more strategy, Nelder-Mead simplex, than CCHHO be. Notwithstanding, the results reveal that CCNMHHO is not only less effective than CCHHO on handling the global optimization and feature selection problems but also higher computational expense than CCHHO because of the extra strategy.

The main contributions of this paper are summarized as followed:

(1) An effective variant of HHO, CCHHO, is presented by introducing the vertical and horizontal crisscross of CSO into the original HHO to handle global optimization, engineering optimization and feature selection problems.
(2) The proposed algorithm utilizes the dimensional mutation of vertical crossover in CSO to avoid the stagnancy in local optima, the horizontal crossover in CSO to reduce the unreachable search spots and strengthen the exploratory capability. It improves the balance between exploration and exploitation capability. Furthermore, the convergence rate is accelerated using competitive operator.
(3) The performance of the proposed CCHHO is investigated by addressing 4 categories of benchmark functions chosen from 23 classical function and CEC2014 and 4 constrained engineering optimization tasks.
(4) The developed CCHHO is adopted to develop a binary feature selection technique for dealing with feature selection problems on 13 datasets.
(5) The proposed CCHHO outperforms CCNMHHO, which is based on the strategies of CSO and Nelder-Mead simplex, in terms of performance and computational expense on the global optimization and feature selection problems.

The remainder of this paper is organized as follows: Sect. 2 explains the background of HHO and CSO and demonstrates the main structure of the proposed algorithm. And Sect. 3 provides the evaluations of the proposed approach using a set of experiments and analysis on global benchmark problems. The practicality of the proposed method in dealing with engineering optimization problems are verified in Sect. 4. Section 5 gives the application of the proposed optimizer on feature selection. Finally, conclusions and future work are drawn in Sect. 6.

2 Literature Study

The advantage of a stochastic optimization algorithm can be determined by whether the algorithm can appropriately harmonize its two significant characteristics: exploration and exploitation [66]. Nevertheless, the randomness of the stochastic optimizers leads to their drawbacks including premature convergence, poor diversity and imbalance between exploratory capability and exploitative capability [9]. Therefore, many studies were devoted to developing a novel excellent algorithm or improving an existing algorithm for solving a variety of optimization problems, which is always a challenging task [67].

As a novel bioinspired algorithm, HHO has been already paid more and more attentions on many application fields.

At the very beginning, HHO was applied in the area of energy efficiency. EHHO [36] was first proposed based on chaotic strategy for closing the best solution and opposition-based theory for exploration. It is investigated on parameter estimation of photovoltaic model such as single diode module (SDM), double diode module (DDM), photovoltaic (PV) model and the manufacture's datasheet. Experimental results demonstrated that EHHO is superior to the well-regarded competitors such as BLPSO, CLPSO, IJAYA and GOTBLBO in terms of not only root mean square error (RMSE) but also convergence speed for identifying the parameters of SDM, DDM and PV model. It also revealed that EHHO can be a beneficial approach to identify the parameters of solar cells in case of some harsh outdoor environment with high irradiance or low temperature. For the same study cases of the PV parameter extraction, Liu et al. [49] proposed an improved HHO, named CCNMHHO, using the Nelder-Mead simplex as well as the horizontal and vertical crossover of the CSO. CCNMHHO employed CSO to improve the population quality by enriching the information exchange between individuals and avoid the local optima by preventing the dimensional stagnation of individuals. Additionally, the Nelder-Mead simplex facilitated enhancing individual searching abilities in aspects of the local search phase and the acceleration of convergence. Experimental results revealed that CCNMHHO was very competitive in estimating the unknown parameters of diverse PV models when comparing to some state-of-the-art methods such as IJAYA, GOTLBO, MLBSA, CPSO, ABSO, ABC, CPSO, EHHO, and so on. And it performed well in solving the complex outdoor environments with diverse temperature and radiance.

HHO has been also applied in satellite image. [39] presented a dynamic HHO (DHHO/M) as a new satellite image segmentation approach, which was incorporated with a dynamic control factor and mutation strategy. The proposed DHHO/M was more efficient to improve the search capability and escape from local optimum than HHO. And experimental results indicated that, in the aspects of fitness function evaluation, image segmentation effect and statistical tests, the DHHO/M-based thresholding approach outperformed original HHO, the advanced multilevel thresholding methods including TLBO, WOA-TH, IDSA and BDE as well as the thresholding approaches based on different criteria such as Tsallis entropy based MGOA, MABC, Otsu-based MFPA and GWO. Furthermore, the practicality and feasibility on real engineering problems of DHHO/M was evaluated using four oil pollution images.

In addition, there are also some classification model optimizations using HHO. In [41], HHO provided a strengthened performance to optimize the artificial neural network (ANN) for improving the accuracy of ANN in predicting slope stability. Results illustrated that HHO raised the prediction accuracy of ANN in terms of RMSE and mean absolute error, and the correlation between actual values of the safety factor and the outputs of HHO-ANN was more significant than ANN. A HHO-CNN hybrid model for classifying hand gesture images was proposed in [44]. After tuning the hyper-parameter of CNN using HHO, the proposed model attained an accuracy of 100% and was superior to the existing models such as WOA-CNN, GSA-HHO, CSA-HHO, PSO-HHO, GA-HHO, ABC-HHO and GWO-HHO.

HHO also can be used in discrete optimization problems. An upgraded binary HHO, HHOSRL, [40] was proposed based on specular reflection learning (SRL) to accurately identify the decisive factors in the early recognition and discrimination of COVID-19 severity. Experimental results showed that the indicators HHOSRL selected, such as age, PaO_2, $SaO_2\%$, Na^+ and LAC, were essential for early accurate estimation of COVID-19 severity. Moreover, HHOSRL performed satisfactorily when fusing with various classifiers and achieved an accuracy of almost 100.0%. The kernel extreme learning machine enabled HHOSRL performed best on blood sample dataset. HHOSRL is the best feature selection method in terms of specificity, sensitivity, accuracy, MCC and time consumption when comparing with other feature selection techniques including bMFO, BGSA, bALO, BSSA, bHHO, bGWO, BPSO, BBA and bWOA. IHHO was proposed in [68], in which salp swarm algorithm was embedded in the update stage of HHO for solving global optimization and feature selection problems. The evaluation results of the proposed IHHO demonstrated that it provided a faster convergence rate and better performance on benchmark function optimization problems than 11 conventional swarm-based algorithms including HHO, DE, GWO, WOA, SSA, MFO, SCA, PSO, MVO, ALO, and GOA as well as 11 state-of-the-art optimizer such as JADE, SaDE, jDE, CoDE, SHADE, ALCPSO, CLPSO, BLPSO, HCLPSO, EPSO and HPSO_TVAC. The IHHO-based feature selection method outperformed BBA, BSSA and BHHO in respect of fitness, feature-length and error rate on ten benchmark feature selection issues. However, from the viewpoint of computational complexity, IHHO may spend much execution time.

As set forth, the own characteristics and wide usage of HHO reflect the outstanding optimization capability. As a consequence, this study proposes a variant of HHO named CCHHO using CSO strategy. However, the CSO mechanism was also introduced in CCNMHHO. It should be noted that there are some differences between the proposed HHO and CCNMHHO. First, in addition to the CSO strategy, CCNMHHO used Nelder-Mead simplex as another mechanism. Second, CCHHO and CCNMHHO are not used in the same application field. The former is employed to handle global optimization, engineering optimization and feature selection problems, while the latter is used as a parameter extraction method of photovoltaic models. We will compare their performances in experiments on global optimization and feature selection problems in later sections.

3 Materials and Methods

3.1 Overview of HHO

HHO is a stochastic swarm intelligence optimizer. This optimizer imitates the collaborative behavior of Harris hawks' swarm in hunting an escaping prey. The mathematical model is mainly based on two explorative phases and four exploitative processes, which are performed randomly.

In HHO, the Harris' hawks represent the search agents; the intended prey indicates the best or approximately optimal solutions obtained so far. In the explorative phases, Harris hawks randomly perch for locating prey. Two explorative phases are modeled based on the positions that hawks roosting. This pattern of exploration boosts the randomness of HHO. Accordingly, it is tending to enable the positions of search agents extend all over possible region of search space. While in the exploitative processes, once detecting the intended prey, the hawks attempt to employ four chasing strategies to cope with various escaping motions of the prey. Four chasing strategies are involved in four exploitative phases. In Appendix C of Supplementary material, Fig. C.1

draws each searching stage in HHO. Meanwhile, the detailed steps including the mathematical model of each stage are described as followed.

Step 1: Initialization.

The initial parameters such as population size N, maximum number of iterations T are determined. And the hawk population (search agents) $\boldsymbol{X}$ is defined randomly.

Step 2: Evaluation.

First, make sure that the current agent explores in the search space. In case of this, the best agent $\boldsymbol{X}_{target}$ is determined after evaluating the fitness of each search agent.

Step 3: Updating the escaping energy.

Due to the reducing escaping energy of the prey, the evolutionary process switches between exploration and exploitation. Obviously, the mutative energy greatly affects the optimization ability of HHO. This time-vary energy, marked as *Escaping_energy*, is defined according to the following equations [26]:

$$Escaping\_energy = 2E_0\left(1 - \frac{t}{\mathrm{T}}\right) \tag{1}$$

$$E_0 = 2rand - 1, \tag{2}$$

where E_0 represents random initial state, which is changed at each generation within the range $(-1, 1)$. When $|Escaping\_energy| \geq 1$, the search agent is updated by the explorative operation in Step 4, otherwise, when $|Escaping\_energy| < 1$, the search agent is modified by the exploitative operation in Step 5.

Step 4: Exploration.

The search agent is distributed in various areas of the search space so as to explore the best agent. According to the equal probability q, the position of the search agent is determined to be a random location when $q \geq 0.5$, while somewhere near other search agents enough when $q < 0.5$. Therefore, there are two explorative phases expressed as Eq. 3 [26].

$$\boldsymbol{X}(t+1) = \begin{cases} \boldsymbol{X}_r(t) - r_1\left|\boldsymbol{X}_r(t) - 2r_2\boldsymbol{X}(t)\right|, q \geq 0.5 \\ \left(\boldsymbol{X}_{target}(t) - \boldsymbol{X}_{average}(t)\right) - r_3 \\ \times\left(r_4(\mathrm{UB} - \mathrm{LB}) + \mathrm{LB}\right), q < 0.5 \end{cases} \tag{3}$$

$$\boldsymbol{X}_{average}(t) = \frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}} X_i(t), \tag{4}$$

where $\boldsymbol{X}_{average}(t)$ is the average position of the current search agents, $\boldsymbol{X}_i(t)$ represents the location of the i-th search agent at generation t. $\boldsymbol{X}(t+1)$ and $\boldsymbol{X}(t)$, respectively, indicate the position of search agents at generation $(t+1)$ and t. $\boldsymbol{X}_r(t)$ denotes the position of a random agent. $\boldsymbol{X}_{target}(t)$ is the position of the best agent. r_1, r_2, r_3, r_4 and q are produced randomly between 0 and 1 at each generation.

Step 5: Exploitation.

According to different escaping status of the prey, four strategies for hawks are used in hunting the prey. Therefore, four exploitative processes are constructed to acquire the search agents around the global optimum produced so far. Besides the escaping energy, a random factor r in $(0, 1)$ is employed to determine which exploitative process is performed. In addition, *Jump_strength* represents the prey's jump strength ought to be updated, it is calculated using the following equation [26]:

$$Jump_{strength} = 2 \times (1 - rand) \tag{5}$$

The description and execution condition of each process is given below. Based on two factors: *Escaping_energy* and r, one sub-step in Step 5.1–5.4 is chosen to complete the exploitation. It is remarkable that the soft besiege demonstrated in Step 5.1–5.2 occurs when $|Escaping\_energy| \geq 0.5$ and the hard besiege described in Step 5.3–5.4 occurs when $|Escaping\_energy| < 0.5$. Moreover, when $r < 0.5$ such as seeing Step 5.2 and Step 5.4, in both soft and hard besiege, the exploitative process can be the movement with progressive rapid dives, which is more intelligent because of the utilization of Levy flight.

If $|Escaping\_energy| \geq 0.5$ and $r \geq 0.5$, perform Step 5.1;
If $|Escaping\_energy| \geq 0.5$ and $r < 0.5$, perform Step 5.2;
If $|Escaping\_energy| < 0.5$ and $r \geq 0.5$, perform Step 5.3;
If $|Escaping\_energy| < 0.5$ and $r < 0.5$, perform Step 5.4;

Step 5.1: The search agents are estimated as followed [26]:

$$\boldsymbol{X}(t+1) = \Delta\boldsymbol{X}(t) - Escaping\_energy \times \left|Jump\_strength \times \boldsymbol{X}_{target}(t) - \boldsymbol{X}(t)\right| \tag{6}$$

$$\Delta\boldsymbol{X}(t) = \boldsymbol{X}_{target}(t) - \boldsymbol{X}(t), \tag{7}$$

where $\Delta\boldsymbol{X}(t)$ is the difference between the position of the best agent and the current search agents.

Step 5.2: The exploitative process can be modeled mathematically such that [26]:

$$\boldsymbol{Y} = \boldsymbol{X}_{target}(t) - Escaping\_energy \times \left|Jump\_strength \times \boldsymbol{X}_{target}(t) - \boldsymbol{X}(t)\right| \tag{8}$$

$$\boldsymbol{Z} = \boldsymbol{Y} + \boldsymbol{S} \times Levy(\mathrm{D}), \tag{9}$$

where $\boldsymbol{S}$ is a random vector in the search space with size of $1 \times \mathrm{D}$ and boundary of $[0, 1]$. Levy flight $Levy(\mathrm{D})$ is used to evaluate the rapid dives [26].

$$Levy(x) = 0.01 \times \frac{u \times \sigma}{|v|^{1/\beta}}, \ \sigma = \left[\frac{\Gamma(1+\beta)\sin(\pi\beta)/2}{\Gamma((1+\beta)/2) \times \beta \times 2^{(\beta-1)/2}} \right]^{1/\beta}, \tag{10}$$

where u, v are random factors and $\beta = 1.5$. Finally, the search agents can choose their next movements using the following rule [26]:

$$\boldsymbol{X}(t+1) = \begin{cases} Y, ifF(\boldsymbol{Y}) < F(\boldsymbol{X}(t)) \\ Z, ifF(\boldsymbol{Z}) < F(\boldsymbol{X}(t)) \end{cases}, \tag{11}$$

where $F(\bullet)$ is the fitness evaluation function.

Step 5.3: The search agents are approximate to the best according to the following equation [26]:

$$\boldsymbol{X}(t+1) = \boldsymbol{X}_{target}(t) - Escaping_{energy} \times |\Delta \boldsymbol{X}(t)|. \tag{12}$$

Step 5.4: In this process, all of the search agents are tended to be closer to the best agent. The search agent can be evaluated according to Eq. 13 [26].

$$\boldsymbol{X}(t+1) = \begin{cases} \boldsymbol{Y}', ifF\left(\boldsymbol{Y}'\right) < F(\boldsymbol{X}(t)) \\ \boldsymbol{Z}', ifF\left(\boldsymbol{Z}'\right) < F(\boldsymbol{X}(t)) \end{cases}, \tag{13}$$

where

$$\boldsymbol{Y}' = \boldsymbol{X}_{target}(t) - Escaping\_energy \times \left| Jump\_strength \times \boldsymbol{X}_{target}(t) - \boldsymbol{X}_{average}(t) \right| \tag{14}$$

$$\boldsymbol{Z}' = \boldsymbol{Y}' + \boldsymbol{S} \times Levy(\mathrm{D}). \tag{15}$$

Step 6: Repeat Steps 2–5 to perform optimization iteratively until the number of iterations has reached the iterative maximum.

Step 7: Obtain the optimal solution ($\boldsymbol{X}_{target}$).

3.2 Overview of CSO

The population-based CSO is an efficient stochastic search optimizer with a crisscross search strategy based on the Confucian doctrine of golden mean. Incorporating a competitive (CP) operator with a dual crossover search behavior, which is different with the crossover in genetic algorithm, CSO algorithm can address the optimization problems, especially complex ones, with high-quality solution and accelerated convergence.

In the optimizing process of CSO, the dual crossover search behavior constructs an interacting chain of two searching operators, horizontal crossover (HC) operator and vertical crossover (VC) operator, respectively. HC is performed based on the capability of social recognition among individuals. While VC operation lies in the capability of each individual's self-recognition. They are alternatively performed with respective probabilities to generate the moderation solutions in each iteration. After each crossover operating, CP operator is performed to estimate the moderation solutions (MS). The MS with better fitness values will be maintained in the new generation. The solutions reproduced in CP operator are referred to as dominant solutions (DS). DS are served as the parent population of the next HC or VC as well as the competitors of the MS stemming from one corresponding crossover operator. It is obvious that the population is modified twice in each iteration by HC and VC, respectively. And CP operator occurs twice to adjust the modified population.

The interacting process based on these operators in CSO can be illustrated in Fig. C.2. Each operator in CSO mentioned above is presented specifically as the following subsections. Suppose that the population X involving N search agents is trained in a D-dimensional search space. Then, the i-th search agent can be represented as $\boldsymbol{X}_i = \left[\boldsymbol{X}_{i,1}, \boldsymbol{X}_{i,2}, \cdots, \boldsymbol{X}_{i,\mathrm{D}}\right]^{\mathrm{T}}, i = \{1, 2, \cdots, \mathrm{N}\}$.

3.2.1 HC operator

HC is a cross-border search mechanism. HC operator as a global optimization operator, randomly divides the population into two parts without repetition at first and then conducts a crossing over arithmetically on two distinct search agents across all their dimensions, which are, respectively, chosen from each subpopulation. After HC operation, each search agent is updated to be MS for HC. The following arithmetical model [50] can be used to express HC operation for a pair of different search agents at the d-th dimension.

$$\begin{cases} \boldsymbol{MSH}_{i,d} = r_1 \times \boldsymbol{X}_{i,d} + (1 - r_1) \times \boldsymbol{X}_{j,d} + c_1 \times (\boldsymbol{X}_{i,d} - \boldsymbol{X}_{j,d}) \\ \boldsymbol{MSH}_{j,d} = r_2 \times \boldsymbol{X}_{j,d} + (1 - r_2) \times \boldsymbol{X}_{i,d} + c_2 \times (\boldsymbol{X}_{j,d} - \boldsymbol{X}_{i,d}) \end{cases}, \tag{16}$$

where $\boldsymbol{MSH}_{i,d}$ and $\boldsymbol{MSH}_{j,d}$ represent the moderation solutions produced by HC, while $X_{i,d}$ and $\boldsymbol{X}_{j,d}$ are their corresponding distinct parent search agents at the d-th dimension, which are employed to perform HC. r_1, r_2, c_1 and c_2 are random parameters that distributed uniformly, where the values of r_1 and r_2 are in $[0, 1]$; and c_1, c_2 are considered as expansion coefficients in the range $[-1, 1]$. The expansive coefficients are very influential in determining the search scope of a search agent.

According to Eq. 16, in HC phase, the multi-dimensional search space is split into two parts. Each part is taken as a hypercube space with the paired parent search agents as their diagonal vertices. As we can see in Eq. 16, each MS contains two components: the first component is originated from the arithmetical crossover in genetic algorithm, which ensures HC searching in each part of the separate hypercubes space with a larger probability; while the second component applies the expansive coefficients, which ensures HC reducing effectively the unreachable blind spots by the first component and searching in the peripheral space of each part with a decreasing probability.

Additionally, to adequately reach the global search ability of HC operator, the HC probability P_h that HC operation occurs is set as 1. Within each generation, a CP operator performs a greedy selection after HC operation. The MS achieved by HC compete with the corresponding parents to produce DS as the new modified population. Accordingly, the survived DS are taken as the parent population of the following VC operation.

3.2.2 VC operator

In VC operation, the information is exchanged between the paired dimensions of a search agent with VC probability P_v. Performed in the opposite direction of HC's horizontal direction, the VC operator randomly splits the dimensions of each parent search agent into two parts without repetition and then operates a crossing over arithmetically on all search agents between two diverse dimensions. The two distinct dimensions are, respectively, selected from each part of dimensions. The parent population is the population of DS derived from HC operation. After VC operation, each search agent is updated to be MS for VC.

Notably, considering that each dimension of search agent exists different units or different boundaries, the normalization and reverse normalization should be performed before and after the VC operation, respectively.

Before VC, the normalization operation on the parent search agents makes sure that the dimension offspring can be produced by VC within the boundary of each dimension. The normalized search agent can be calculated as below [50]:

$$\boldsymbol{NX}_{i,j} = \frac{\boldsymbol{X}_{i,j} - \mathrm{LB}_j}{\mathrm{UB}_j - \mathrm{LB}_j}, \tag{17}$$

where $\boldsymbol{NX}_{i,j}$ represents the i-th normalized search agent at the j-th dimension. LB and UB are the minimum and maximum of the i-th search agent X_i at the j-th dimension, respectively.

VC operation on the paired dimensions of the i-th search agent can be shown as follows [50]:

$$\boldsymbol{NV}_{i,d1} = r \times \boldsymbol{NX}_{i,d1} + (1 - r) \times \boldsymbol{NX}_{i,d2}, d1, d2 \in (1, \mathrm{D}), \tag{18}$$

where $\boldsymbol{NV}_{i,d1}$ indicates the moderation solution with normalized form generated by VC, while $\boldsymbol{NX}_{i,d1}$ and $\boldsymbol{NX}_{i,d2}$ are, respectively, corresponding parent search agents at the $d1$-th and $d2$-th dimension, which are adopted to execute VC. r is uniformly distributed parameter whose value is randomly generated in the range [0, 1].

After VC, the reverse normalization operation can be expressed by the following equation [50]:

$$\boldsymbol{MSV}_{i,j} = \boldsymbol{NV}_{i,j} \times \left(\mathrm{UB}_j - \mathrm{LB}_j\right) - \mathrm{LB}_j, \tag{19}$$

where $\boldsymbol{MSV}_{i,j}$ represents the moderated solution produced by VC operator.

From Eq. 18, VC search happens between two diverse dimensions of a single search agent and generates a single offspring. This exchange of dimensional information dramatically provides an opportunity to avoid dimensional stagnancy in population, hence it makes the stagnant dimension escape from local optimum without destroying other dimensions that may be the global optimum.

Since certain dimensions of the population may simultaneously encounter the premature convergence in optimization process, the VC probability P_v, which determines the number of the paired dimensions taking part in VC operation, is set in [0.2, 0.8] [50].

Similar to HC, after VC search operation, CP operation is also performed to generate DS as the new offspring population of the current population. It means that the outperformed DS are regarded as the parent population of HC search operation in the next generation.

3.2.3 CP Operator

As aforementioned, CP operator is applied to choose the better search agent after the population is updated by each crossover search. It plays a vital role in enhancing the search quality. CP operator employs a greedy selection mechanism to update the search agents, which can be expressed as below [50]:

$$\boldsymbol{DS} = \begin{cases} X, & \mathit{if}\ f(\boldsymbol{X})\ \mathit{is\ better\ than}\ f(MS) \\ MS, & \mathit{if}\ f(\boldsymbol{MS})\ \mathit{is\ better\ than}\ f(X) \end{cases}, \tag{20}$$

where DS represents the new modified population for the next crossover operation, which are the better solutions with better fitness between MS generated by HC or VC and its parent $\boldsymbol{X}$.

From the arithmetical model of CP operator, only that MS who are superior to their parents can be survival, otherwise, they would be dropped out of the competition. The CP mechanism can simply make the search agents maintain the better solutions and converge to the global solution rapidly.

3.3 The Proposed CCHHO

In this subsection, with the aid of the separate inherent excellent abilities of the three operators in CSO, we construct an enhanced optimizer, referred as CCHHO, by incorporating

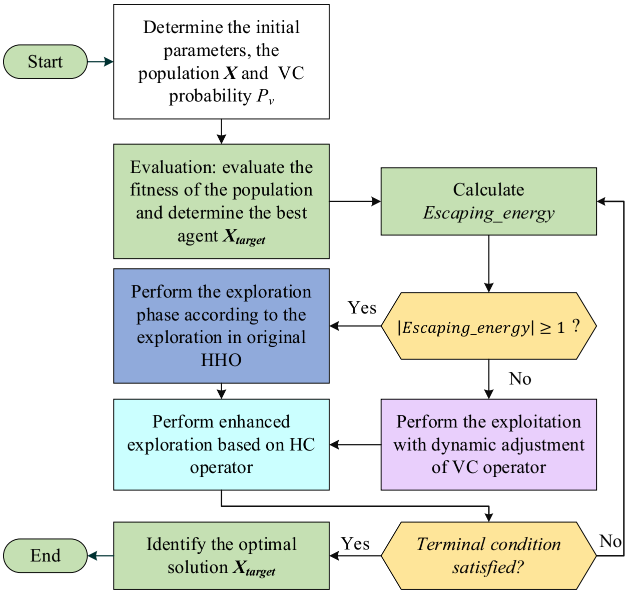
Start
Determine the initial parameters, the population X and VC probability P_v
Evaluation: evaluate the fitness of the population and determine the best agent X_{target}
Calculate Escaping_energy
|Escaping_energy| ≥ 1 ?
Yes
No
Perform the exploration phase according to the exploration in original HHO
Perform the exploitation with dynamic adjustment of VC operator
Perform enhanced exploration based on HC operator
Terminal condition satisfied?
Yes
No
Identify the optimal solution X_{target}
End

Fig. 1 Structure of the proposed CCHHO

the operators into HHO. The structure of the proposed CCHHO is illustrated in Fig. 1, and its corresponding pseudocode is described in Appendix A of Supplementary material. The proposed CCHHO optimizer contains three phases: exploration, exploitation with dynamic adjustment, enhanced exploration. Similar to the evolutionary process of each individual in the conventional HHO, in each generation, the escaping energy determines to perform the first exploration phase or the second exploitation phase. While the third phase optimizes all of the individuals produced from the first two phases in the current generation once more.

Exploration is the original exploration of HHO, which is divided into two stochastic exploring means based on the equal probability q. The search agents are randomly explored or searched around other search agents. This randomness is contributed to search the global optimal solution in an extensive search space. In this phase, a new exploring population is generated.

Exploitation with dynamic adjustment is to adjust the exploitation of the conventional HHO adaptively using a VC regulatory mechanism. The current search agent is modified through one of the four exploitation stages in the conventional HHO and recorded for the subsequent competition. Considering the possibility of premature convergence caused by the Levy flight in the preceding exploitation stages of HHO, VC search mechanism is employed to provide a dimensional mutation dynamically. The dimensional mutation is tended to make the stagnant dimension of a search agent jump out of the local optimum so as to boost the other stagnant dimensions to jump out of the local optimum as soon as possible. VC as an adjustment mechanism is performed in this phase with a certain vertical crossover probability after the preceding exploitation stages. As the architecture of VC search mechanism described in Fig. 2, the dimensions of each current search agent are split into two equal parts. And then a crossover is operated on paired dimensions. There is only part of dimension of the agent updated in resulting search agent. Similar to the VC in CSO, the CP operation is used to compete the resulting search agent with the recorded search agent. An adjusted population consists of the survival search agents. It is obvious that this VC mechanism integrated with CP operator can not only eliminate the local optimality caused by the dimensional stagnancy in previous exploitation stages but also achieve an improved stability between diversification and intensification.

Enhanced exploration is to perform HC search mechanism on all search agents as an enhanced global search operator. This phase is performed after updating the whole population in the first two phases. Therefore, the exploring or adjusted population originated from the above two phases is regarded as the parent population of HC operator. As demonstrated in Fig. 3, the population is split into two equal subpopulations. And a search agent is, respectively, chosen from each of the subpopulations. Each paired search agents are crossed

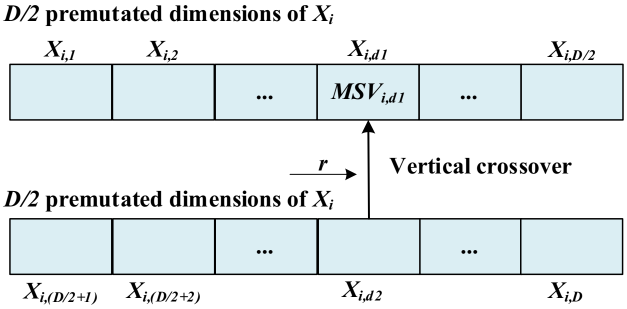

Fig. 2 The architecture of VC search mechanism

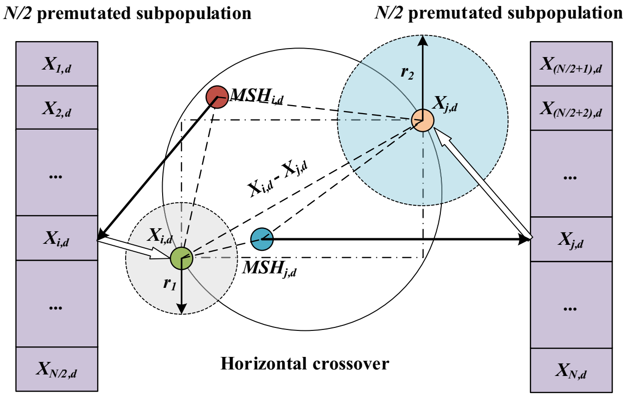

Fig. 3 The architecture of HC search mechanism

based on HC mechanism. The same as HC in CSO, the CP operator is also introduced to perform a greedy selection between the resulting population of HC and its parent population. And then, the competitive individuals are addressed as the parent population of the next generation. In this phase, HC mechanism effectively facilitates to not only achieve the better solutions in different search spaces but also decrease the inaccessible scotoma, which cannot be obtained in the above two phases, in the solution space so as to boost the ability to search the global optimum. In this case, the excellent explorative ability of HC can fill the gap in the exploration of the conventional HHO. The enhanced exploration with HC mechanism improves the quality of the solution.

Moreover, we can see that the VC and HC with the CP operator introduced in the suitable phases of the evolutionary process of the proposed CCHHO can explore the high-quality global best solution for a faster convergence and a diversified population.

The complexity of CCHHO can be calculated based on the following five processes: initialization, exploration phase of HHO, the vertical crossover, horizontal crossover and competitive operator. First, for N search individuals, the computational complexity of initialization is$\boldsymbol{O}(\mathrm{N})$; Secondly, the computational complexity of exploration phase is$\boldsymbol{O}(\mathrm{T} \times \mathrm{N} \times \mathrm{D})$; third, the computational complexity of vertical crossover is$\frac{1}{2}\boldsymbol{O}(\mathrm{T} \times \mathrm{N} \times \mathrm{D})$; Fourth, the computational complexity of horizontal crossover is$\frac{1}{2}\boldsymbol{O}(\mathrm{T} \times \mathrm{N} \times \mathrm{D})$; finally, the competitive operator conducted after each crossover mechanism is$2\boldsymbol{O}(\mathrm{T} \times \mathrm{N})$. Therefore, the total computational complexity of CCHHO is$2\boldsymbol{O}(\mathrm{N} \times (\mathrm{T} \times \mathrm{D} + \mathrm{T} + 1))$, in which T is the maximum number of iterations and D is the dimension of the problem. It can be seen that the introduction of crisscross strategy into HHO makes the computational time increase one time, but does not rise the computational complexity.

4 Evaluations of the Proposed CCHHO

In this section, a series of experiments were conducted to investigate the efficiency and strength of incorporated mechanisms in the proposed CCHHO algorithm in a variety of cases. For all approaches in each experimental case, we recorded the fair results due to utilizing an identical experimental setup such as operating environment and parameter settings. All experiments are executed on Windows 10 (64 bit) operating system with Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40 GHz 2.39 GHz (two processors) and 8GBs RAM. The development tool MATLAB R2017a is used to code and run all algorithms. Considering evaluating the effectiveness of CCHHO optimizer from a more comprehensive perspective, the following experiments are conducted.

Experiment 1: Evaluating the performance of CCHHO on benchmark function optimization using four categories of benchmark functions. The tested functions contains three unimodal, three multimodal and three fixed-dimension multimodal test functions, which are chosen from 23 classical functions [69], as well as three composition functions available from IEEE2014 [70] test problems. The CCHHO is compared with 9 well-regarded bioinspired algorithms including HHO and CSO as well as 9 state-of-the-art optimizers including CCNMHHO on above 12 test functions. In addition, we perform the analysis of CCHHO' performance with different dimension, different population size and different maximum iteration number.

Experiment 2: Evaluating the performance of CCHHO on engineering optimization problems such as tension/compression spring design, welded beam design and pressure vessel design. The optimization performance of CCHHO on these problems are compared with some existing engineering optimizer as well as CCNMHHO.

Experiment 3: Evaluating the CCHHO's capability of tackling the feature selection problems on 13 UCI datasets. CCHHO as a feature selection approach is compared with 6 feature selection methods also including CCNMHHO.

4.1 Experiments on Global Optimization Problems

CCHHO is compared with 9 well-regarded optimizers and 9 state-of-the-art algorithms on 12 test functions, which are of different categories, including unimodal (F1–F3), multimodal (F4–F6), fixed-dimension multimodal (F7–F9) and composition (F10–F12) test functions. In Appendix B of Supplementary material, Table B.1 and Table B.2 show the mathematical descriptions of these benchmark functions. According to the own characteristics of each type of the problems, these functions were used to validate various properties of the proposed CCHHO algorithm. The unimodal and multimodal sets are suitable for examining the performance of the tested approach in terms of convergency, exploitative capability and explorative capability. Whereas, the rest test sets of CEC 2014 test functions were applied to evaluate the overall abilities of the tested algorithm to equilibrize exploration and exploitation.

In each experimental case, every chosen approach runs 30 times independently aiming at weakening the impact of randomness on experimental results. Also, the population size is set to 30. In computational intelligence, a matter of fact is the fairness of computational testing [71]. According this rule, we can make sure the obtained results are justifiably recorded without bias toward one or other algorithm [72]. The setup of initial parameters of all algorithms was identical with their original references. The setup of the parameters in all the comparing algorithms are recorded in Table1.

Table 1 Parameter settings for all the comparing algorithms

Algorithm	Parameters
CCHHO	$c_1, c_2 \in [-1, 1]; \varepsilon_1, \varepsilon_2, \varepsilon \in [0, 1]; p_1 = 1; p_2 = 0.6$
DE	$F_{min} = 0.2; F_{max} = 0.8; CR = 0.1$
PSO	$c1 = 2; c2 = 2; vMax = 6$
SCA	$a = 2$
GWO	$a = [2, 0]$
MFO	$b = 1; t = [-1, 1]; a \in [-1, -2]$
WOA	$a_1 = [2, 0]; a_2 = [-2, -1]; b = 1$
BA	$a = 0.5; r = 0.5$
GSA	$G_0 = 100; \alpha = 20$
CSO	$c_1, c_2 \in [-1, 1]; \varepsilon_1, \varepsilon_2, \varepsilon \in [0, 1]; p_1 = 1; p_2 = 0.6$
HHO	$RabbitEnergy = [2, 0]$
LSHADE_cnEpSin	$\mu F = \mu CR = \mu freq = 0.5; H = 5; freq = 0.5; ps = 0.5; pc = 0.4$
SaDE	$F_1 = 0.9, C_{r1} = 0.1; F_2 = 0.9, C_{r2} = 0.9; F_3 = 0.5, C_{r3} = 0.3;$ $F_4 = 0.5, C_{r4} = 0.3; F_5 = 0.5, C_{r5} = 0.3$
LSHADE	$r^{N^{init}} = 18; r^{arc} = 2.6; p = 0.11; H = 6$
CGPSO	$c1 = c2 = 1.49618; w = 0.7289$
CLPSO	$w = [0.2, 0.9]; c = 1.496$
ALCPSO	$c_1 = c_2 = 2.0; w = 0.4; \Theta_0 = 60; \mathrm{T} = 2; pro = \frac{1}{\mathrm{D}};$ $V_{max} = 0.5 * searchrange$
ACWOA	$a_1 = [2, 0]; a_2 = [-2, -1]; b = 1$
IWOA	$b = 1; \mathrm{crossover} = 0.1$
CCNMHHO	$c_1, c_2 \in [-1, 1]; \varepsilon_1, \varepsilon_2, \varepsilon \in [0, 1]; p_1 = 1; p_2 = 1$

In this study, to estimate the optimization capability of the comparative algorithms, the statistical performance measures such as standard deviation and mean, which can be used to show the robustness of the examined algorithm, are employed to record the experimental results. In the table recording the experimental results, the "mean" and "std" label represent, respectively, the average value and standard deviation of 30 independent runs for each function. Moreover, we employed several extensive statistical significance results to evaluate the success of CCHHO. The Wilcoxon signed rank test at the significant level of 0.05 was applied to analyze statistically the significant difference of statistical measures among the comparative methods. In each Wilcoxon signed rank test result, the results of the significant analysis are shown in the description identified by "+/=/−", in which "+" represents the number of functions that CCHHO is superior to other tested methods significantly, whereas, "−" represents the number of functions that CCHHO is worse than others significantly, "=" represents the number of functions that the significant difference does not exist between CCHHO and other comparing methods. For further statistical analysis, we employed the Friedman test to present the average ranking performance of each tested method more evidently. The "ARV" label represents the average performance ranking of the algorithm based on Friedman test for overall benchmark functions. The "rank" label represents the overall ranking of each algorithm. The best results are bolded in the experimental results.

4.1.1 Comparison with Well-regarded Bioinspired Algorithms

In this subsection, nine well-regarded bioinspired algorithms were used to assess the efficiency of the proposed CCHHO. This comprehensive group of algorithms contains DE [19], PSO [8], SCA [20], GWO [21], MFO [22], WOA [23], BA [24], GSA [25] and CSO [50]. In this experimental case, the maximum number of function evaluations is set to 30,000. Table B.3 expresses the statistical outcomes of comparing the developed CCHHO with the selected methods on tackling 12 benchmark functions. The p-values of Wilcoxon signed rank test analyzing the significant difference between paired methods are presented in Table B.4. Table 2 provides the significant statistical results from Wilcoxon signed-rank test and Friedman test.

From the outcomes of each methods listed in Table B.3 and Table B.4, it is obviously that CCHHO can achieve the best solutions on all of the functions. For F1–F3, the optimal solutions of F1, F2 are able to be found by CCHHO. While other competitors including CSO cannot obtain the optima. It indicates that CCHHO inherits the excellent exploitative

Table 2 Comparing results between CCHHO and bioinspired algorithms over 12 test functions

Metric	CCHHO	DE	PSO	SCA	GWO	MFO
+/=/−	−/−/−	11/0/1	12/0/0	12/0/0	11/1/0	12/0/0
ARV	1.5292	4.1931	7.8278	7.7000	4.5056	6.7347
Rank	1	3	10	9	4	6
Metric	~	WOA	BA	GSA	CSO	
+/=/-	~	11/0/1	12/0/0	12/0/0	7/4/1	
ARV	~	6.0042	6.9139	7.2583	2.3333	
Rank	~	5	7	8	2	

ability of original HHO, CSO and other compared optimizers are prone to premature convergence. For F4–F5, CCHHO outperforms other competitors except CSO. There are no significant difference between the performances CCHHO and CSO on F4–F6. This reveals that CCHHO gets the good explorative ability of CSO to escape from local optima. For F7–F9, CCHHO performs worse than CSO and DE on F7, equal with CSO on F9. However, CCHHO also can get the best solution on each function. For F10–F12, CCHHO is significantly superior to other comparing algorithms including CSO, which shows that CCHHO can achieve a better equilibrium between exploration and exploitation than others.

Observing from Table 2, the overall statistical outcomes of significance in Wilcoxon signed rank test demonstrate that CCHHO produces 7 significantly better/4 equal/1 significantly worse when compared with CSO, which is the worst case. And CCHHO performs significantly worse than DE, WOA and CSO on only 1 function, equal with CSO and GWO on, respectively, 4 functions and 1 function. This indicates that CCHHO has overwhelmingly advantages over other approaches. It makes sense that CCHHO achieved the best ARV of 1.5292 in the Friedman test, which outperforms CSO in second place with ranking value of 2.3333. Therefore, we can draw the conclusion that the developed CCHHO was the best method with considerable success over 9 well-established bioinspired algorithms.

The curves in Fig.C.3 intuitively shows the convergence rate of CCHHO, DE, PSO, SCA, GWO, MFO, WOA, BA, GSA and CSO on addressing eight functions. As we can detected from Fig.C.3, CCHHO converges faster with the best solution than other competitors on all functions except F9. CCHHO achieves a competitive convergence rate on F9. It approaches the best solutions later than CSO, but it attains the best solution, which is not significantly different with the solution obtained by CSO. For other competitors, their accelerated trends are tend to stagnate into local optimum during early evolutionary stage, whereas CCHHO can converge fastest with the high-quality solutions on these optimization tasks. Therefore, based on the enhanced exploratory trends of HC operator, the convergence speed of CCHHO is also boosted. It is obvious that these tendencies can confirm the enhancement of CCHHO in most cases of the unimodal, multimodal and composition problems.

In summary, these overall results on each type of problems verify that the proposed HHO has a steadily efficient evolutionary capability at an accelerated convergence rate. This indicates that, in CCHHO, the HC mechanism facilitates strengthening the global search capability; the reasonable adjustments of HC and VC mechanisms are able to successfully make an excellent balance between exploitation and exploration trends; and the CP operator is helpful to boost the convergence rate.

4.1.2 Comparison with State-of-the-Art Algorithms

In this subsection, the advantages of the proposed CCHHO were investigated by comparing against several state-of-the-art optimization algorithms such as LSHADE_cnEpSin [58], SaDE [59], LSHADE[60], CGPSO [61], CLPSO [62], ALCPSO [63], ACWOA [64], IWOA [65], CCNMHHO [49]. In this experimental case, the maximum number of function evaluations is set to 45,000. Table B.5

Table 3 Comparing results between CCHHO and state-of-the-art algorithms over 12 test functions

Metric	CCHHO	LSHADE_cnEpSin	SaDE	LSHADE	CGPSO	CLPSO
+/=/-	−/−/−	11/1/0	9/2/1	10/2/0	12/0/0	9/3/0
ARV	2.2861	5.8014	4.9319	5.7625	6.3736	5.8236
Rank	1	5	3	4	8	6
Metric	~	ALCPSO	ACWOA	IWOA	CCNMHHO	
+/=/-	~	10/2/0	8/4/0	12/0/0	8/4/0	
ARV	~	6.6514	6.1167	7.3583	3.8944	
Rank	~	9	7	10	2	

records the relevant mathematical results in terms of the statistical metrics, Wilcoxon statistical test and Friedman test. The p values in Table B.6 demonstrate the statistical significance at 5% degree of the Wilcoxon test between the proposed CCHHO and each competitor. Table 3 provides the significant statistical results from Wilcoxon signed-rank test and Friedman test. Figure C.4 visibly illustrates the aforementioned results in terms of the convergence rate.

As observed in Table B.5, CCHHO can exhibit the best performance on all twelve functions except F8. On unimodal cases, the CCHHO has an excellent ability of exploitative search due to finding the optimal solutions on F1, F2 and the best solution on F3, while the ACWOA, as the second top exploitative algorithm on unimodal problems, achieves the optimal solution on F1 and F2. But ACWOA cannot attain a satisfactory solution on F3. Moreover, CCNMHHO is less effective than CCHHO, which indicates that the exploitative ability of CCNMHHO is decreased resulting from the introduction of Nelder-mead simplex. On multimodal cases, the CCHHO obtains the best solutions on all multimodal functions, while the CLPSO and CCNMHHO can achieve the best solutions only on F5 and F6. This observation reveals that the enhanced explorative ability of CCHHO outperforms other methods. In other words, the integrating the Nelder-mead simplex may not boost the explorative capability of CCNMHHO. On the fixed-dimension cases, the performance of CCHHO is competitive to other comparing algorithms. CCHHO obtains the best solution on F7, which is not significantly different with LSHADE_cnEpSin, SaDE, LSHADE, CLPSO, ALCPSO and CCNMHHO. Moreover, according to the standard deviation, CCHHO is more stable than CCNMHHO and ALCPSO. For F8, CCHHO performs worse than SaDE, but it also can attain the best solution. For F9, CCHHO is superior to all other algorithms. For the composition functions, CCHHO is significantly superior to other splendid algorithms including CCNMHHO. As we can see that the coordination between exploration and exploitation of CCHHO is the best of that of all comparing algorithms.

According to the significant statistical results in Table 3, CCHHO performs 11/9/10/12/9/10/8/12/8 significantly better, 1/2/2/0/3/2/4/0/4 equal, 0/1/0/0/0/0/0/0/0 significantly worse when comparing with LSHADE_cnEpSin, SaDE, LSHADE, CGPSO, CLPSO, ALCPSO, ACWOA, IWOA, CCNMHHO. Accordingly, from each perspective, the performance of CCHHO outperforms the selected outstanding competitors. Although the ACWOA also has a nice exploitive capability and SaDE and CLPSO can perform a relatively good explorative search, these are only single advantage for them. Furthermore, the results of Friedman test reveal that CCHHO ranks the first among ten algorithms, followed by CCNMHHO, SaDE, LSHADE, LSHADE_cnEpSin, CLPSO, ACWOA, CGPSO, ALCPSO, IWOA. Therefore, although CCNMHHO can achieve an excellent performance, the Nelder-mead simplex in this method weakens its exploitative capability, and then makes the coordinate between exploration and exploitation worse than CCHHO.

Figure C.4 exposes the above merits of the proposed CCHHO by means of convergence curves. According to the curves, we detect that the convergence rate of CCHHO is accelerated on several functions including F1, F3, F4, F5, F11. It is also obvious that most of other advanced competitors occurs premature convergence on F1, F3, F4, F5, F11, F12. For F7, F9, F12, CGPSO traps into the local optimum with faster convergence than CCHHO, which can converge to the best solutions. These trends indicate that the convergence rate of CCHHO is accelerated in searching the global solution. Notably, the CCHHO has a faster convergence speed than CCNMHHO on each function. It confirms that CCHHO can effectively harmonizes the high search ability and accelerated convergence rate.

The numerical results and convergence curves indicate that CCHHO outperforms the other comparing state-of-the-art optimizers with higher convergence rate. Therefore, it makes sense that the HC and VC mechanisms of CSO are contributed to enhancing the CCHHO's performance comprehensively and effectively.

Accordingly, the comprehensive effectiveness of the proposed CCHHO is the best among all comparing optimizers. First, CCHHO can provide high explorative capability due to combing the horizontal crossover into the later evolutionary stage of HHO. Second, CCHHO has strong exploitation ability resulting from the reasonable integration of the excellent exploitative ability of HHO's levy flight and the adjustment of vertical crossover. Third, CCHHO attains an accelerated convergence speed. Finally, the experimental results verify that the proposed CCHHO has a suitable coordination between exploration and exploitation with the help of the CSO strategy. Hence, CCHHO has a better effectiveness than CCNMHHO, which is also improved using CSO strategy, on coping with the global optimization problems. In other words, employing more mechanisms may not enhance the optimization performance of the algorithm. In the next subsections, the effectiveness of CCHHO will be verified in some more challenging real problems such as engineering optimization and feature selection.

4.1.3 Scalability Evaluation

Above subsections presented the merits of the proposed CCHHO by comparing with other well-established and advanced methods. This section provides a scalability evaluation to analyze the impact of dimensions of the optimization task on the efficiency of CCHHO. In this evaluation, the CCHHO algorithm was compared against the original HHO on six benchmark functions with increasing dimensions of 100, 200, 500, 1000 and 2000, respectively. Table B.7 reports the statistical results for each dimension of six optimization cases tackled by CCHHO and HHO.

As Table B.7 demonstrated, the comparison results on each dimension are very stable to show the superiority of the CCHHO to HHO. On each dimension, the proposed CCHHO can attain the same optimal or best solutions as HHO when addressing all functions. However, the CCHHO significantly outperforms HHO on the same magnitude when tackling F3, F5, F6. Hence, observing from the standard deviations, CCHHO is significantly more stable than HHO on any dimension. For F1, F2, F4 and F6, both of CCHHO and HHO achieve the optimal or best solution whatever the dimension is. It means that the optimization ability of the proposed CCHHO is enhanced on the problems in different dimensions due to the combination of two crisscross operations. And the success of the proposed method is not affected by the increasing dimensions of the optimization problems. In addition, it can be known that CCHHO produces the best performance when dimension is 2000, and the worst on dimension of 500.

4.1.4 Sensitivity Analysis

Since majority of metaheuristic algorithms perform on a general idea, rather than provide particular domain knowledge for each problem, diverse values of parameters set for an algorithm may result in diverse performance [73]. It is necessary to identify the parameter settings for algorithm and then can provide robustness and applicability in handling a variety of problems in different scientific areas. The performance of the proposed CCHHO is sensitive to the initial parameter such as the population size (N) and maximum number of iterations (T). A sensitivity analysis is done to examine the impact of these two parameters on CCHHO. For the sensitivity analysis, N is set to $\{30, 50, 80, 100\}$ and T is set to $\{100, 500, 800, 1000\}$. The analysis is conducted using these parameters on functions selected from four categories of unimodal, multimodal, fixed-dimension multimodal and composition test functions in Table B.1 and Table B.2. They are, respectively, F3, F5, F8 and F10. For each function, the maximum number of function evaluation is 30000. The sensitivity analyzing results are shown in aspects of mean and convergence rate for the four functions.

First, from the perspective of population size (N), CCHHO is simulated for different population size and fixed number of iterations. The iteration number is fixed to 1000. Table B.8 gives the average fitness values of CCHHO on F3, F5, F8, F10 with four different population size. It reveals that CCHHO obtains the best performance with a small population size. Additionally, from the convergence curves of CCHHO on F5 related with different population size in Fig. C. 5(a), the sensitivity of CCHHO decreases with population size.

Second, from the perspective of maximum number of iterations (T), CCHHO is executed for different number of iterations. Table B.9 recorded the average fitness values of CCHHO on F3, F5, F8, F10 with four different number of iterations. Figure C. 5(b) presents the convergence curves of CCHHO on F5 with different number of iterations. It can be detected from Table B.9 and Fig.C. 5(b) that CCHHO converges to optimal solution with the increasing of the number of iterations. It shows that the iterations number is essential to the convergence and robustness of CCHHO.

4.2 Applications to Engineering Optimization Problem

The reliability of the proposed CCHHO can be assessed on solving the real-world engineering optimization problems. In this subsection, CCHHO is employed to tackle three well-known engineering optimization tasks, which are tension/compression spring design [74], welded beam design [74] and pressure vessel design [74, 75] problem. The outcomes of optimization for solving each engineering problem were compared to a variety of standard and advanced methods proposed in previous literature.

4.2.1 Tension/Compression Spring Design Problem

The objective of the tension/compression spring design problem is to minimize the weight of a spring by determine the optima of three structural variables such as wire diameter (d), the number of active coils (N) and mean coil diameter (D). The mathematical definition of this case is as follows [74]:

$$\text{Consider} \quad \vec{x} = \left[x_1 x_2 x_3\right] = [dDN], \text{ Min.} \quad f(\vec{x}) = \left(x_3 + 2\right)x_2 x_1^2, \tag{21}$$

Table 4 Overall results of tension/compression spring design problem

Optimizer	Optimal variables			Optimal weight
	d	N	D	
OBLGOA [3]	0.0530178	9.6001616	0.38953229	0.01270136
BWOA [76]	0.051602	11.2441198	0.357488	0.0126654
IFFOA [77]	0.051657	11.334999	0.355939	0.0126655
IGWO [78]	0.051701	11.275600	0.356983	0.0126670
PSOCSCALF [79]	0.0518836	11.018738	0.36141614	0.012665923
CMSSA [79]	0.051789	11.14985	0.359116	0.0126655
RW-GWO [80]	0.05167	11.33056	0.35613	0.012674
RGWO [9]	0.0514017	11.7065	0.349818	0.012704
FCMHMD [74]	0.0516917	11.285123	0.3567832	**0.01266523**
HHO[26]	0.051796393	11.138859	0.359305355	0.012665443
CCNMHHO	0.051633912	11.36708866	0.355392446	0.012665288
CCHHO	0.05168488	11.29486632	0.35661714	**0.01266523**

$$\text{Subject to}\quad g_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0,$$

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566\left(x_2x_1^3 - x_1^4\right)} + \frac{1}{5108x_1^2} \le 0,$$

$$g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0,$$

$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \le 0$$

Variables range $0.05 \le x_1 \le 2, 0.25 \le x_2 \le 1.30, 2.00 \le x_3 \le 15$

The tension/compression spring design problem is one of the most ordinary engineering optimization problems. It has already been employed to verify the effectiveness of the new optimizers in many studies. The proposed CCHHO is compared with several recent well-established methods including OBLGOA, BWOA, IFFOA, IGWO, PSOCSCALF, CMSSA, RW-GWO, RGWO, FCMHMD, HHO and CCNMHHO. CCNMHHO is coded for this problem. The results shown in Table 4 point out that the proposed CCHHO and FCMHMD can find the optimal variables for this problem with the weight of 0.1266523. The CCNMHHO, BWOA, HHO, IFFOA, CMSSA, which achieve competitive results, are respectively, ranked in the 2nd, 3rd, 4th, 5th and 6th places.

4.2.2 Welded Beam Design Problem

The main aim of this well-known case is to design a welded beam to determine four decision variables that minimize the fabrication cost. The decision variables include the weld thickness (h), length of an attached part of the bar (l), height of the bar (t) and thickness of the bar (b). The mathematical definition of this case can be expressed by the following [74]:

$$\text{Consider}\quad \vec{x} = \left[x_1x_2x_3x_4\right] = [hltb],$$

$$\text{Min.}\quad f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4\left(14.0 + x_2\right), \tag{22}$$

$$\text{Subject to}\quad g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \le 0,$$

$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \le 0,$$

$$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \le 0,$$

$$g_4(\vec{x}) = x_1 - x_4 \le 0,$$

$$g_5\left(\vec{\vec{x}}\right) = P - P_c(\vec{x}) \le 0,$$

$$g_6(\vec{x}) = 0.125 - x_1 \le 0,$$

$$g_7(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4\left(14.0 + x_2\right) - 5.0 \le 0$$

Variables range $0.1 \le x_1, x_4 \le 2, 0.1 \le x_2, x_3 \le 10$

$$\text{Where}\quad \tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},$$

$$\tau' = \frac{P}{\sqrt{2x_1x_2}},$$

$$\tau'' = \frac{MR}{J},\quad M = P\left(L + \frac{x_2}{2}\right), R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},$$

$$J == 2\left\{\sqrt{2x_1x_2}\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\},$$

$$\sigma(\vec{x}) = \frac{6PL}{x_4x_3^2},\quad \delta(\vec{x}) = \frac{6PL^3}{Ex_4x_3^2},$$

$$P_c(\vec{x}) = \frac{\sqrt[4.013E]{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),$$

$$P = 6000lb,\quad L = 14in., \delta_{\max} = 0.25in.,$$

$$\tau_{\max} = 13,600psi, \sigma_{\max} = 30,000psi,$$

$$E == 30 \times 10^6 psi, G = 12 \times 10^6 psi$$

The welded beam design problem has been extensively handling by various optimizers such as BSAISA, BWOA,

Table 5 Overall results of welded beam design problem

Optimizer	Optimal variables				Optimal cost
	h	l	t	b	
BSAISA [81]	0.205730	3.470489	9.036624	0.205730	1.724852
BWOA [76]	0.205829	3.251922	9.034556	0.205829	1.695620
OBLGOA [3]	0.205769	3.471135	9.032728	0.2059072	1.7257
CMSSA [82]	0.205113	3.264134	9.036661	0.205729	1.695838
GDFA [83]	0.205670	3.472000	9.036680	0.205730	1.724700
CGWO [84]	0.343891	1.883570	9.03133	0.212121	1.725450
VCS [85]	0.20572964	3.47048867	9.03662391	0.20572964	1.72485231
C-ITGO [86]	0.2057296	3.4704886	9.0366239	0.2057296	1.7248523
IAHLO [87]	0.205730	3.470489	9.033624	0.205730	1.724852
SFO [66]	0.2038	3.6630	9.0506	0.2064	1.73231
SFS [88]	0.20572964	3.47048867	9.03662391	0.20572964	1.72485231
HHO[26]	0.204039	3.531061	9.027463	0.206147	1.73199057
FCMHMD [74]	0.2056756	3.4700936	9.043817	0.205693	1.725684
SSC [89]	0.1992	3.4307	9.1045	0.2051	1.7222
RGWO [9]	0.20569	3.4718	9.0365	0.20574	1.7253
CCNMHHO	0.20571349	3.253420735	9.036575827	0.205731928	1.69527319
CCHHO	0.20572195	3.253272321	9.03656748	0.20573221	1.69526617

OBLGOA, CMSSA, GDFA, CGWO, VCS, C-ITGO, IAHLO, SFO, SFS, HHO, FCMHMD, SSC and RGWO. To evaluate the ability of the proposed CCHHO in addressing this engineering design case, the estimated variables and optimal cost obtained by this optimizer are compared against those of these optimizers and CCNMHHO. The detailed comparison results listed in Table 5 reveal that the proposed CCHHO algorithm is able to identify the best design for this engineering case by attaining the optimal cost of 1.69526617. This cost is the lower than that of other algorithms. CCNMHHO finds the optimal cost of 1.69527319, which is higher than CCHHO.

4.2.3 Pressure Vessel Design Problem

The main purpose of pressure vessel design problem is to design the cylindrical pressure vessel with a minimum total cost, which contains the cost of materials and welding. To tackle this case, four parameters (T_s, T_h, R, L) must be optimized. T_s and T_h denote the thickness of the shell and head, respectively; R is the inner radius; L represents the length of the cylindrical section of the vessel without head. The mathematical model of this case can be defined as follows [74]:

$$\begin{aligned}&\text{Consider}\quad \vec{x} = \left[x_1 x_2 x_3 x_4\right] = \left[T_s T_h R L\right],\\&\text{Min.}\quad f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7881x_2x_3^2\\&+ 3.1661x_1^2x_4 + 19.84x_1^2x_3,\end{aligned} \tag{23}$$

Table 6 Overall results of pressure vessel design problem

Optimizer	Optimal variables				Optimal cost
	T_h	T_s	R	L	
OBLGOA [3]	0.81622	0.40350	42.291138	174.811191	5966.67160
RW-GWO[80]	0.81250	0.43750	42.09840	176.63784	6059.736
OBSCA[90]	1.2500	0.0625	59.1593	70.8437	5833.9892
MHDA[91]	0.778169	0.384649	40.3196	200	5885.3353
WOA [23]	0 0.812500	0 0.437500	42 0.0982699	176 0.638998	6059 0.7410
HHO [26]	0.81758383	0.4072927	42.09174576	176.7196352	6000.46259
FCGHMD[74]	0.8129	0.401571	42.087015	176.85321	5952.89502
CCNMHHO	0.942465189	0.46586103	48.8323932	107.8873575	5833.620039
CCHHO	0.94877596	0.46898044	49.15937475	105.15766782	5826.47696325

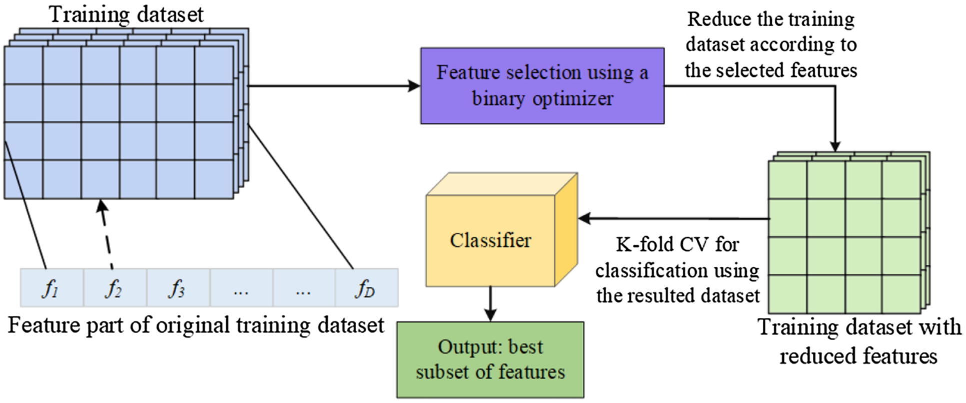
Training dataset
Feature selection using a binary optimizer
Reduce the training dataset according to the selected features
f_1
f_2
f_3
...
...
f_D
Feature part of original training dataset
Classifier
K-fold CV for classification using the resulted dataset
Output: best subset of features
Training dataset with reduced features

Fig. 4 Architecture of FS based on a binary optimizer

Subject to $g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0$,

$$g_2(\vec{x}) = -x_{12} + 0.00954x_3 \leq 0,$$

$$g_3(\vec{x}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0,$$

$$g_4(\vec{x}) = x_4 - 240 \leq 0$$

Variables range $0 \leq x_1, x_2 \leq 99, 10 \leq x_3, x_4 \leq 200$

Some of optimization algorithms previously utilized to tackle the pressure vessel design problem include OBLGOA, RW-GWO, OBSCA, MHDA, WOA, HHO and FCGHMD. The performance of the CCHHO algorithm on optimizing this design problem is compared with that of these algorithms and CCNMHHO. The comparison outcomes in Table 6 report the optimal cost of 5826.47696325. It reveals meaningfully that the CCHHO is the most effective optimizer among above well-established ones in this engineering design case.

According to the analysis of the above three constrained engineering optimization tasks, the proposed CCHHO has confirmed its efficiency in addressing real-world engineering optimization problems over other well-regarded optimizers. It indicates that the superiority of CCHHO results from the HC and VC operators that interact reasonably with the HHO mechanism. Also, due to the constrained property and uncertain search space in the engineering optimization problems, we can see that CCHHO can perform optimization in the search domain with infeasible spaces.

4.3 Application to Feature Selection

Feature selection (FS), as an optimization process of dimensionality reduction, is an essential stage of addressing the high-dimensional search space in classification problem. In this section, the proposed CCHHO is used in dealing with this challenging process for further assessing the availability of CCHHO in real-word application. FS problem aims at determining a subset with the fewest representative features from the overall set of properties in original dataset to achieve an optimal classification accuracy. Therefore, as the architecture of FS based on a binary optimizer illustrated in Fig. 4, it is observed that the classification accuracy of the classifier can be considered as the measurement of validating the ability of dimensionality reduction.

4.3.1 FS Based on Binary CCHHO

The FS problem can be taken as a discrete optimization problem. Therefore, the proposed CCHHO is transformed into a binary version, called BCCHHO, since the solved FS task is characterized as a binary optimization problem. Hence, in the implementing process of CCHHO for solving FS problem, the continuous solutions are mapped into discrete forms with binary value. The solution ought to be represented as a binary vector with the same dimensions as the number of features in the training dataset. In the vector with binary values, the selected attribute is labelled with value 1 at its corresponding location, whereas the non-selected attribute is marked with value 0.

First, we produced binary values in accordance with random thresholds to initialize the population as followed:

$$x_{i,j} = \begin{cases} 0 & rand \leq 0.5 \\ 1 & rand > 0.5 \end{cases}, \tag{24}$$

where $x_{i,j}$ denotes the binary value in the location vector of search agent at i-th row and j-th column.

Second, without influencing the architecture of CCHHO, we can employ the transfer function to map the continuous vector of search agent into binary form in the formal transformation. In this study, we chose the S-shaped transfer function [92], shown as Eq. 25, to perform the mapping to squash the continuous vectors in each dimension.

$$S_{Trans} = \frac{1}{1 + e^{\left(-x/3\right)}}, \tag{25}$$

Table 7 Comparison results of the proposed BCCHHO vs. other optimizers in aspect of fitness

Dataset	BCCHHO	BHHO	bGWO	BBA	bWOA	BSSA	BCCNMHHO
ARV	**1.9385**	6.4846	2.4308	6.5077	4.0308	4.6115	1.9962
Rank	1	6	3	7	4	5	2

Table 8 Comparison results of the proposed BCCHHO vs. other optimizers in aspect of error rate

Dataset	BCCHHO	BHHO	bGWO	BBA	bWOA	BSSA	BCCNMHHO
ARV	**2.2538**	5.7308	2.3462	7	4.2346	4.0500	2.3846
Rank	1	6	2	7	5	4	3

Table 9 Comparison results of the proposed BCCHHO vs. other optimizers in aspect of feature-length

Dataset	BCCHHO	BHHO	bGWO	BBA	bWOA	BSSA	BCCNMHHO
ARV	2.3923	5.7077	3.5654	5.2423	2.8731	5.8615	2.3577
Rank	2	6	4	5	3	7	1

Table 10 Comparison results of the proposed BCCHHO vs. other optimizers in aspect of computational time

Dataset	BCCHHO	BHHO	bGWO	BBA	bWOA	BSSA	BCCNMHHO
ARV	5.8462	5.1538	2.3000	3.1308	1.6077	2.9615	7
Rank	6	5	2	4	1	3	7

where $\boldsymbol{x}$ represents the continuous vector of search agent. ***S_Trans*** denotes a continuous intermediate form as the output of the transfer function. To achieve a binary vector, ***S_Trans*** is changed and compared with the initial binary vector generating in the initialization based on the following equation [68]:

$$x_{i,j} = \begin{cases} outputPos = {}^{\sim}initialPos \;\; rand \leq \boldsymbol{S\_Trans} \\ outputPos = initialPos \;\;\; rand > \boldsymbol{S\_Trans} \end{cases}, \tag{26}$$

where *initialPos* is the initial binary value, *outputPos* indicates the new binary value attained.

Finally, we applied the classifier to evaluate the selected subset of features obtained from BCCHHO. FS is evidently regarded as a multi-objectives problem since it involves acquiring the highest classification accuracy with fewest feature subsets. Therefore, considering the objectives comprehensively, we can evaluate the search agent in accordance with the classification accuracy and the number of chosen features using the fitness function as follows [68]:

$$fitness = \alpha \times \left(N_s/\mathrm{N}\right) + \beta \times (1 - accuracy), \tag{27}$$

where N_s denotes the number of features filtered by BCCHHO. N is the total number of features in original dataset. *accuracy* represents the classification accuracy calculated from the classifier. $1 - accuracy$ indicates the error rate. α and β are considered as the weights of the selected features' number and classification accuracy, respectively, $\alpha \in [0, 1]$, $\beta = 1 - \alpha$. They represent the importance of the selected features' number and error rate.

4.3.2 Experimental Setup

We present a comparative study to examine the optimization behavior of the binary CCHHO compared to several state-of-the-art bioinspired algorithms including BHHO (binary HHO [26]), bGWO [93], BBA [94], bWOA (binary WOA [23]), BSSA (binary salp swarm algorithm [95]) and BCCNMHHO, which is the binary versions of CCNMHHO [49] improved using CSO and Nelder-mead simplex. Thirteen practical benchmark datasets on distinct subject areas were utilized as case studies. They are available from the UCI repository (https://archive.ics.uci.edu/ml//datasets.php, available at October 2022) and the Wielaw dataset is from the literature [96]. As presented in Table B.10, in these datasets, the sizes of instances and features are distinguished with each other. That is beneficial to assess the proposed method from distinct perspectives.

To alleviate the bias of feature selection, in the training of classification, k-fold cross-validation (CV) is employed to assess the optimality of each selected feature subset. The dataset in k-fold CV was segmented into *k* equivalent parts. One part in the dataset is used as the testing set to estimate the classifier's classification accuracy, while the *k*–1 parts is the training set for training the classifier. The evaluation metrics of FS are the average values obtained from k validations for each problem.

For fairness in the investigation, each evaluation associated with the BCCHHO is performed in the same computational environment. The initial parameters of all comparison methods were identical with that in their original references. Furthermore, for each involved method, the search agents and maximum iteration are, respectively, set as 20 and 50. It runs ten independent times. And the common KNN was used as classifier in training. The fold k in k-fold CV was set as 10, and α is 0.05 [68, 97].

We adopt four evaluation metrics to express the experimental results. They are the average fitness, average error rate, average number of the selected features and average computational time. Moreover, the corresponding standard deviation (std) is accompanied in the results to estimate the performances of the investigated FS approaches on thirteen datasets on ten independent runs. According to the selected features subsets in the evaluated datasets, the average fitness and average error rate are attained. Additionally, the average ranking value (ARV) in aspects of the evaluation metrics based on the Friedman test are employed to detect the optimal of solutions among the examined optimizers. The best values of four evaluation metrics are bold in the results.

4.3.3 Results and Discussion

The simulation results attained by BCCHHO against other competitors are recorded in Table B.11, 12, 13, 14 in aspects of average fitness, error rate, numbers of selected features as well as computational time, respectively. Tables 7, 8, 9, 10 demonstrate the average ranking results of all investigated optimizers based on Friedman Test.

Observing the results of average fitness in Table B.11, BCCHHO shows competitive outcomes of average fitness. The proposed optimizer can attain the best average fitness on 7 of 13 datasets, while BCCNMHHO and bGWO get the best solution on 5 and 4 datasets, respectively. However, the average fitness obtained by BCCHHO are very close to the best value. Considering Table B.11, 12, 13, BCCHHO can achieve the lowest error rate with relatively fewer features on Exactly, M-of-n, WineEW, Zoo, vehicle, wdbc and Wielaw. On other datasets, bGWO and BCCNMHHO can obtain the best fitness. Nevertheless, the lowest error rates are mostly attained by bGWO at the cost of selecting more features. And BCCNMHHO cannot realize the lowest error rates, which indicates that BCCNMHHO have missed the important information that affect the accuracy because of selecting the fewest features. This also can be concluded from Tables 7, 8, 9. According the average ranking value based on Friedman test, BCCHHO is ranked first in terms of average fitness and error rate. BCCNMHHO ranks in the first position in terms of feature-length, but it cannot outperform BCCHHO in terms of the two important metrics such as fitness and error rate.

Inspecting the computational time shown in Table B.14 and Table 10, BCCNMHHO expends the most running time among the tested algorithms, followed by BCCHHO, BHHO. However, as the variants of the original HHO, BCCNMHHO takes more than three times as long as that of HHO. It can be known that the time expend of BCCNMHHO is influenced by the introduced Nelder-mead simplex and CSO strategies. The computational expend of BCCHHO is the second most expensive, nevertheless, it is close to that of HHO and less half of BCCNMHHO's time cost. Moreover, the algorithms associated with HHO spends more time than others. It indicates that the time cost of the modified algorithm depends on not only the introduced mechanism but also its original algorithm.

As summary, the BCCHHO can achieve the best fitness and the lowest error rate with relatively fewer features on the datasets with low-dimensional as well as high-dimensional datasets. This indicates that the proposed BCCHHO provides a remarkable success among the tested FS techniques. It can be obviously concluded that the proposed BCCHHO considerably overcomes the limitations of the original version of BHHO by integrating the HC and VC operators. This optimizer is suite for the discrete feature selection problem. Additionally, the above analysis of the computational time reveals that the computational time of an algorithm is affected by its mechanism of finding optimum.

5 Conclusions and Future Directions

In this study, an enhanced HHO (CCHHO) optimizer was proposed, in which the horizontal crossover strategy in CSO is fused with the update stage of HHO, the vertical crossover strategy in CSO is introduced into the exploitation phase of HHO and the competitive operator in CSO is also performed after each crossover. The combination of CSO in HHO ensures not only a good balance between local search capability and global search ability but also an accelerated convergence speed. The CCHHO's performance is comprehensively assessed on four categories of functions. The results show that CCHHO is more efficient than other well-known bioinspired algorithms and advanced optimizers in terms of convergence and the balance between exploitation and exploration. Simultaneously, on solving the engineering optimization problems, the CCHHO outperforms other previous optimizers to lower manufacturing costs. Hence, for the FS problems, binary CCHHO achieves higher accuracy by selecting fewer features than other competitors. In a word, CCHHO can handle continuous global optimization problems and discrete feature selection with an outstanding effectiveness. Additionally, we detect that CCHHO is superior to CCNMHHO on solving the above problems. It indicates that the Nelder-mead simplex introduced in CCNMHHO may not be contributed to achieving the best efficiency when addressing these problems, while it increases the algorithm's computational time. Consequently, it is not always the case that more strategies added into an algorithm can make it perform better. Moreover, an algorithm cannot be a universal method for solving every problem, which is proven by "NFL" theorem.

Accordingly, we can attempt to evaluate what other problems the CCHHO can work out. In this study, we adopt the CCHHO in single objective problems and low-dimensional feature selection. As a future study, the CCHHO can be used for multi-objective optimization problems and high-dimensional feature selection such as gene selection. Additionally, due to its verified excellent capability of optimization and applicability in feature selection, we can also employ it to enhance the classifier's accuracy by optimizing its key parameters and simultaneously selecting the optimal feature subsets.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s42235-022-00298-7.

Availability of Data and Materials The data involved in this study are all public data, which can be downloaded through public channels.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Gupta, S., & Deep, K. (2019). Improved sine cosine algorithm with crossover scheme for global optimization. *Knowledge Based Syst, 165*, 374–406. https://doi.org/10.1016/j.knosys.2018.12.008
2. Gharehchopogh, F. S., & Gholizadeh, H. (2019). A comprehensive survey: whale optimization algorithm and its applications. *Swarm Evolution Comput, 48*, 1–24. https://doi.org/10.1016/j.swevo.2019.03.004
3. Ewees, A. A., Abd Elaziz, M., & Houssein, E. H. (2018). Improved grasshopper optimization algorithm using opposition-based learning. *Expert Systems with Applications, 112*, 156–172. https://doi.org/10.1016/j.eswa.2018.06.023
4. Long, W., Jiao, J. J., Xu, M., Tang, M. Z., Wu, T. B., & Cai, S. H. (2022). Lens-imaging learning Harris hawks optimizer for global optimization and its application to feature selection. *Expert Systems with Applications, 202*, 117255. https://doi.org/10.1016/j.eswa.2022.117255
5. Beheshti, Z. (2022). BMPA-TVSinV: a binary marine predators algorithm using time-varying sine and V-shaped transfer functions for wrapper-based feature selection. *Knowledge Based Syst, 252*, 109446. https://doi.org/10.1016/j.knosys.2022.109446
6. Zhu, S. K., Wu, Q., Jiang, Y. X., & Xing, W. (2021). A novel multi-objective group teaching optimization algorithm and its application to engineering design. *Comput Indust Eng, 155*, 107198. https://doi.org/10.1016/j.cie.2021.107198
7. Preeti, & Deep, K. (2022). A random walk Grey wolf optimizer based on dispersion factor for feature selection on chronic disease prediction. *Expert Systems with Applications, 206*, 117864. https://doi.org/10.1016/j.eswa.2022.117864
8. Pramanik, R., Sarkar, S., & Sarkar, R. (2022). An adaptive and altruistic PSO-based deep feature selection method for Pneumonia detection from chest X-rays. *Applied Soft Computing, 128*, 109464. https://doi.org/10.1016/j.asoc.2022.109464
9. Banaie-Dezfouli, M., Nadimi-Shahraki, M. H., & Beheshti, Z. (2021). R-GWO: representative-based grey wolf optimizer for solving engineering problems. *Applied Soft Computing, 106*, 107328. https://doi.org/10.1016/j.asoc.2021.107328
10. Xu, Y. T., Chen, H. L., Luo, J., Zhang, Q., Jiao, S., & Zhang, X. Q. (2019). Enhanced moth-flame optimizer with mutation strategy for global optimization. *Information Sciences, 492*, 181–203. https://doi.org/10.1016/j.ins.2019.04.022
11. Tian, X., & Li, J. (2019). A novel improved fruit fly optimization algorithm for aerodynamic shape design optimization. *Knowledge Based Syst, 179*, 77–91. https://doi.org/10.1016/j.knosys.2019.05.005
12. Arora, S., & Anand, P. (2019). Binary butterfly optimization approaches for feature selection. *Expert Systems with Applications, 116*, 147–160. https://doi.org/10.1016/j.eswa.2018.08.051
13. Jadhav, A. N., & Gomathi, N. (2018). WGC: Hybridization of exponential grey wolf optimizer with whale optimization for data clustering. *Alexand Eng J, 57*, 1569–1584. https://doi.org/10.1016/j.aej.2017.04.013
14. Abbassi, R., Abbassi, A., Heidari, A. A., & Mirjalili, S. (2019). An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models. *Energy Convers Manag, 179*, 362–372. https://doi.org/10.1016/j.enconman.2018.10.069
15. Ateya, A. A., Muthanna, A., Vybornova, A., Algarni, A. D., Abuarqoub, A., Koucheryavy, Y., & Koucheryavy, A. (2019). Chaotic salp swarm algorithm for SDN multi-controller networks. *Eng Sci Tech Intern J, 22*, 1001–1012. https://doi.org/10.1016/j.jestch.2018.12.015
16. Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I., Al-Zoubi, A. M., Mirjalili, S., & Fujita, H. (2018). An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems. *Knowledge Based Syst, 154*, 43–67. https://doi.org/10.1016/j.knosys.2018.05.009
17. Syed, M. A., & Syed, R. (2022). Weighted Salp Swarm Algorithm and its applications towards optimal sensor deployment. *J King Saud Univ Comput Inform Sci, 34*, 1285–1295. https://doi.org/10.1016/j.jksuci.2019.07.005
18. Yang, B., Zhong, L. N., Zhang, X. S., Shu, H. C., Yu, T., Li, H. F., & Sun, L. M. (2019). Novel bio-inspired memetic salp swarm algorithm and application to MPPT for PV systems considering partial shading condition. *Journal of Cleaner Production, 215*, 1203–1222. https://doi.org/10.1016/j.jclepro.2019.01.150
19. Kitayama, S., Arakawa, M., & Yamazaki, K. (2011). Differential evolution as the global optimization technique and its application to structural optimization. *Applied Soft Computing, 11*, 3792–3803. https://doi.org/10.1016/j.asoc.2011.02.012
20. Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. *Knowledge Based Syst, 96*, 120–133. https://doi.org/10.1016/j.knosys.2015.12.022
21. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advanc Eng Software, 69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
22. Mirjalili, S. (2015). Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowledge Based Syst, 89*, 228–249. https://doi.org/10.1016/j.knosys.2015.07.006

23. Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advanc Eng Software, 95*, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
24. Yang, X. S., & Hossein Gandomi, A. (2012). Bat algorithm: A novel approach for global engineering optimization. *Engineering Computations, 29*, 464–483. https://doi.org/10.1108/02644401211235834
25. Wu, P., Wang, H., Li, B., Fu, W., Ren, J., & He, Q. (2022). Disassembly sequence planning and application using simplified discrete gravitational search algorithm for equipment maintenance in hydropower station. *Expert Systems with Applications, 208*, 118046. https://doi.org/10.1016/j.eswa.2022.118046
26. Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: algorithm and applications. *Future Generation Computer Syst, 97*, 849–872. https://doi.org/10.1016/j.future.2019.02.028
27. Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: a new method for stochastic optimization. *Future Generation Computer Syst, 111*, 300–323. https://doi.org/10.1016/j.future.2020.03.055
28. Zhong, C. T., Li, G., & Meng, Z. (2022). Beluga whale optimization: a novel nature-inspired metaheuristic algorithm. *Knowledge Based Syst, 251*, 109215. https://doi.org/10.1016/j.knosys.2022.109215
29. Zhao, S. J., Zhang, T. R., Ma, S. L., & Chen, M. (2022). Dandelion optimizer: a nature-inspired metaheuristic algorithm for engineering applications. *Engineering Applications of Artificial Intelligence, 114*, 105075. https://doi.org/10.1016/j.engappai.2022.105075
30. Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine predators algorithm: a nature-inspired metaheuristic. *Expert Systems with Applications, 152*, 113377. https://doi.org/10.1016/j.eswa.2020.113377
31. Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation, 12*, 702–713. https://doi.org/10.1109/TEVC.2008.919004
32. Gandomi, A. H., Yang, X.-S., & Alavi, A. H. (2011). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering Computations, 29*, 17–35. https://doi.org/10.1007/s00366-011-0241-y
33. Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer Aided Design, 43*, 303–315. https://doi.org/10.1016/j.cad.2010.12.015
34. Yang, X.-S., Karamanoglu, M., & He, X. (2013). Flower pollination algorithm: a novel approach for multiobjective optimization. *Eng Optimizat, 46*, 1222–1237. https://doi.org/10.1080/0305215x.2013.832237
35. Gandomi, A. H., Yang, X.-S., & Alavi, A. H. (2011). Mixed variable structural optimization using Firefly Algorithm. *Computers & Structures, 89*, 2325–2336. https://doi.org/10.1016/j.compstruc.2011.08.002
36. Chen, H. L., Jiao, S., Wang, M. J., Heidari, A. A., & Zhao, X. H. (2020). Parameters identification of photovoltaic cells and modules using diversification-enriched Harris hawks optimization with chaotic drifts. *Journal of Cleaner Production, 244*, 118778. https://doi.org/10.1016/j.jclepro.2019.118778
37. Song, S., Wang, P., Heidari, A. A., Zhao, X., & Chen, H. (2022). Adaptive Harris hawks optimization with persistent trigonometric differences for photovoltaic model parameter extraction. *Eng Appl Artific Intell, 109*, 104608. https://doi.org/10.1016/j.engappai.2021.104608
38. Abdel Aleem, S. H. E., Zobaa, A. F., Balci, M. E., & Ismael, S. M. (2019). Harmonic overloading minimization of frequency-dependent components in harmonics polluted distribution systems using harris hawks optimization algorithm. *IEEE Access, 7*, 100824–100837. https://doi.org/10.1109/access.2019.2930831
39. Jia, H., Lang, C., Oliva, D., Song, W., & Peng, X. (2019). Dynamic Harris Hawks optimization with mutation mechanism for satellite image segmentation. *Remote Sensing, 11*, 1421. https://doi.org/10.3390/rs11121421
40. Hu, J., Han, Z., Heidari, A. A., Shou, Y., Ye, H., Wang, L., & Wu, P. (2022). Detection of COVID-19 severity using blood gas analysis parameters and Harris hawks optimized extreme learning machine. *Comput Biology Med, 142*, 105166. https://doi.org/10.1016/j.compbiomed.2021.105166
41. Moayedi, H., Osouli, A., Nguyen, H., & Rashid, A. S. A. (2021). A novel Harris hawks' optimization and k-fold cross-validation predicting slope stability. *Engineering Computations, 37*, 369–379. https://doi.org/10.1007/s00366-019-00828-8
42. Ramachandran, M., Mirjalili, S., Nazari-Heris, M., Parvathysankar, D. S., Sundaram, A., & Charles Gnanakkan, C. A. R. (2022). A hybrid grasshopper optimization algorithm and harris hawks optimizer for combined heat and power economic dispatch problem. *Engineering Applications of Artificial Intelligence, 111*, 104753. https://doi.org/10.1016/j.engappai.2022.104753
43. Issa, M., & Samn, A. (2022). Passive vehicle suspension system optimization using Harris Hawk optimization algorithm. *Mathematics and Computers in Simulation, 191*, 328–345. https://doi.org/10.1016/j.matcom.2021.08.016
44. Gadekallu, T. R., Srivastava, G., Liyanage, M. M. I., Chowdhary, C. L., Koppu, S., & Maddikunta, P. K. R. (2022). Hand gesture recognition based on a Harris Hawks optimized convolution neural network. *Computers and Electrical Engineering, 100*, 107836. https://doi.org/10.1016/j.compeleceng.2022.107836
45. Suresh, T., Brijet, Z., & Blesslin Sheeba, T. (2021). CMVHHO-DKMLC: a chaotic multi verse Harris Hawks optimization (CMV-HHO) algorithm based deep kernel optimized machine learning classifier for medical diagnosis. *Biomed Sig Proc Control, 70*, 103034. https://doi.org/10.1016/j.bspc.2021.103034
46. Jangir, P., Heidari, A. A., & Chen, H. (2021). Elitist non-dominated sorting Harris hawks optimization: framework and developments for multi-objective problems. *Expert Systems with Applications, 186*, 115747. https://doi.org/10.1016/j.eswa.2021.115747
47. Balaha, H. M., El-Gendy, E. M., & Saafan, M. M. (2021). CovH2SD: a COVID-19 detection approach based on Harris Hawks Optimization and stacked deep learning. *Expert Systems with Applications, 186*, 115805. https://doi.org/10.1016/j.eswa.2021.115805
48. Kamboj, V. K., Nandi, A., Bhadoria, A., & Sehgal, S. (2020). An intensify Harris Hawks optimizer for numerical and engineering optimization problems. *Applied Soft Computing, 89*, 106018. https://doi.org/10.1016/j.asoc.2019.106018
49. Liu, Y., Chong, G. S., Heidari, A. A., Chen, H. L., Liang, G. X., Ye, X. J., & Wang, M. (2020). Horizontal and vertical crossover of Harris hawk optimizer with Nelder-Mead simplex for parameter estimation of photovoltaic models. *Energy Conversion and Management, 223*, 113211. https://doi.org/10.1016/j.enconman.2020.113211
50. Meng, A.-B., Chen, Y.-C., Yin, H., & Chen, S.-Z. (2014). Crisscross optimization algorithm and its application. *Knowledge Based Syst, 67*, 218–229. https://doi.org/10.1016/j.knosys.2014.05.004
51. Meng, A., Chen, S., Ou, Z., Ding, W., Zhou, H., Fan, J., & Yin, H. (2022). A hybrid deep learning architecture for wind power prediction based on bi-attention mechanism and crisscross optimization. *Energy, 238*, 121795. https://doi.org/10.1016/j.energy.2021.121795
52. Weng, S., Tan, W., Ou, B., & Pan, J.-S. (2021). Reversible data hiding method for multi-histogram point selection based on

improved crisscross optimization algorithm. *Information Sciences, 549*, 13–33. https://doi.org/10.1016/j.ins.2020.10.063

53. Meng, A., Zeng, C., Wang, P., Chen, D., Zhou, T., Zheng, X., & Yin, H. (2021). A high-performance crisscross search based grey wolf optimizer for solving optimal power flow problem. *Energy, 225*, 120211. https://doi.org/10.1016/j.energy.2021.120211
54. Kumar, M., & Dhillon, J. S. (2019). A conglomerated ion-motion and crisscross search optimizer for electric power load dispatch. *Applied Soft Computing, 83*, 105641. https://doi.org/10.1016/j.asoc.2019.105641
55. Yin, H., Wu, F., Meng, X., Lin, Y., Fan, J., & Meng, A. (2020). Crisscross optimization based short-term hydrothermal generation scheduling with cascaded reservoirs. *Energy, 203*, 117822. https://doi.org/10.1016/j.energy.2020.117822
56. Kaur, M., Dhillon, J. S., & Kothari, D. P. (2020). Crisscross differential evolution algorithm for constrained hydrothermal scheduling. *Applied Soft Computing, 93*, 106393. https://doi.org/10.1016/j.asoc.2020.106393
57. Patwal, R. S., & Narang, N. (2018). Crisscross PSO algorithm for multi-objective generation scheduling of pumped storage hydrothermal system incorporating solar units. *Energy Conversion and Management, 169*, 238–254. https://doi.org/10.1016/j.enconman.2018.05.067
58. Awad, N. H., Ali, M. Z., & Suganthan, P. N. (2017). Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving CEC2017 benchmark problems. IEEE congress on evolutionary computation (CEC). *Donostia Spain*. https://doi.org/10.1109/CEC.2017.7969336
59. Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation, 13*, 398–417. https://doi.org/10.1109/tevc.2008.927706
60. Tanabe, R., & Fukunaga, A. S. (2014). Improving the search performance of SHADE using linear population size reduction. 2014 IEEE congress on evolutionary computation (CEC). *Beijing China, 2014*, 1658–1665. https://doi.org/10.1109/CEC.2014.6900380
61. Jia, D., Zheng, G., Qu, B., & Khan, M. K. (2011). A hybrid particle swarm optimization algorithm for high-dimensional problems. *Computers & Industrial Engineering, 61*, 1117–1122. https://doi.org/10.1016/j.cie.2011.06.024
62. Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation, 10*, 281–295. https://doi.org/10.1109/tevc.2005.857610
63. Chen, W.-N., Zhang, J., Lin, Y., Chen, N., Zhan, Z.-H., Chung, H.S.-H., & Shi, Y.-H. (2013). Particle swarm optimization with an aging leader and challengers. *IEEE Transactions on Evolutionary Computation, 17*, 241–258. https://doi.org/10.1109/tevc.2011.2173577
64. Elhosseini, M. A., Haikal, A. Y., Badawy, M., & Khashan, N. (2019). Biped robot stability based on an A-C parametric whale optimization algorithm. *Journal of Computational Science, 31*, 17–32. https://doi.org/10.1016/j.jocs.2018.12.005
65. Tubishat, M., Abushariah, M. A. M., Idris, N., & Aljarah, I. (2019). Improved whale optimization algorithm for feature selection in Arabic sentiment analysis. *Applied Intelligence, 49*, 1688–1707. https://doi.org/10.1007/s10489-018-1334-8
66. Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). The sailfish optimizer: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Engineering Applications of Artificial Intelligence, 80*, 20–34. https://doi.org/10.1016/j.engappai.2019.01.001
67. Chen, H., Jiao, S., Heidari, A. A., Wang, M., Chen, X., & Zhao, X. (2019). An opposition-based sine cosine approach with local search for parameter estimation of photovoltaic models. *Energy Conversion and Management, 195*, 927–942. https://doi.org/10.1016/j.enconman.2019.05.057
68. Zhang, Y., Liu, R., Wang, X., Chen, H., & Li, C. (2021). Boosted binary Harris hawks optimizer and feature selection. *Engineering Computations, 37*, 3741–3770. https://doi.org/10.1007/s00366-020-01028-5
69. Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation, 3*, 82–102. https://doi.org/10.1109/4235.771163
70. Liang JJ, Qu BY, Suganthan PN (2013) Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore.
71. Shi, K., Wang, J., Zhong, S., Tang, Y., & Cheng, J. (2020). Non-fragile memory filtering of T-S fuzzy delayed neural networks based on switched fuzzy sampled-data control. *Fuzzy Sets and Systems, 394*, 40–64. https://doi.org/10.1016/j.fss.2019.09.001
72. Zhang, H., Wang, Z., Chen, W., Heidari, A. A., Wang, M., Zhao, X., & Zhang, X. (2021). Ensemble mutation-driven salp swarm algorithm with restart mechanism: framework and fundamental analysis. *Expert Systems with Applications, 165*, 113897. https://doi.org/10.1016/j.eswa.2020.113897
73. Braik, M. S. (2021). Chameleon swarm algorithm: a bio-inspired optimizer for solving engineering design problems. *Expert Systems with Applications, 174*, 114685. https://doi.org/10.1016/j.eswa.2021.114685
74. Abd Elaziz, M., Yousri, D., & Mirjalili, S. (2021). A hybrid Harris hawks-moth-flame optimization algorithm including fractional-order chaos maps and evolutionary population dynamics. *Advances in Engineering Software, 154*, 102973. https://doi.org/10.1016/j.advengsoft.2021.102973
75. Kannan, B. K., & Kramer, S. N. (1994). An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design, 116*, 405–411. https://doi.org/10.1115/1.2919393
76. Chen, H., Xu, Y., Wang, M., & Zhao, X. (2019). A balanced whale optimization algorithm for constrained engineering design problems. *Applied Mathematical Modelling, 71*, 45–59. https://doi.org/10.1016/j.apm.2019.02.004
77. Wu, L., Liu, Q., Tian, X., Zhang, J., & Xiao, W. (2018). A new improved fruit fly optimization algorithm IAFOA and its application to solve engineering optimization problems. *Knowledge-Based Systems, 144*, 153–173. https://doi.org/10.1016/j.knosys.2017.12.031
78. Long, W., Jiao, J., Liang, X., & Tang, M. (2018). Inspired grey wolf optimizer for solving large-scale function optimization problems. *Applied Mathematical Modelling, 60*, 112–126. https://doi.org/10.1016/j.apm.2018.03.005
79. Chegini, S. N., Bagheri, A., & Najafi, F. (2018). PSOSCALF: A new hybrid PSO based on sine cosine Algorithm and Levy flight for solving optimization problems. *Applied Soft Computing, 73*, 697–726. https://doi.org/10.1016/j.asoc.2018.09.019
80. Gupta, S., & Deep, K. (2019). A novel random walk grey wolf optimizer. *Swarm and Evolutionary Computation, 44*, 101–112. https://doi.org/10.1016/j.swevo.2018.01.001
81. Wang, H., Hu, Z., Sun, Y., Qinghua, Su., & Xia, X. (2018). Modified backtracking search optimization algorithm inspired by simulated annealing for constrained engineering optimization problems. *Computation Intellig Neurosci, 2018*, 1–27. https://doi.org/10.1155/2018/9167414

82. Zhang, Q., Chen, H., Heidari, A. A., Zhao, X., Xu, Y., Wang, P., & Li, C. (2019). Chaos-induced and mutation-driven schemes boosting salp chains-inspired optimizers. *IEEE Access, 7*, 31243–31261. https://doi.org/10.1109/access.2019.2902306
83. Wang, C.-F., & Song, W.-X. (2019). A novel firefly algorithm based on gender difference and its convergence. *Applied Soft Computing, 80*, 107–124. https://doi.org/10.1016/j.asoc.2019.03.010
84. Kohli, M., & Arora, S. (2018). Chaotic grey wolf optimization algorithm for constrained optimization problems. *Journal of Computational Design and Engineering, 5*, 458–472. https://doi.org/10.1016/j.jcde.2017.02.005
85. Li, M. D., Zhao, H., WeiWeng, X., & Han, T. (2016). A novel nature-inspired algorithm for optimization: virus colony search. *Advances in Engineering Software, 92*, 65–88. https://doi.org/10.1016/j.advengsoft.2015.11.004
86. Ferreira, M. P., Rocha, M. L., Silva Neto, A. J., & Sacco, W. F. (2018). A constrained ITGO heuristic applied to engineering optimization. *Expert Systems with Applications, 110*, 106–124. https://doi.org/10.1016/j.eswa.2018.05.027
87. Wang, L., Pei, J., Wen, Y., Pi, J., Fei, M., & Pardalos, P. M. (2018). An improved adaptive human learning algorithm for engineering optimization. *Applied Soft Computing, 71*, 894–904. https://doi.org/10.1016/j.asoc.2018.07.051
88. Salimi, H. (2015). Stochastic fractal search: a powerful metaheuristic algorithm. *Knowledge-Based Systems, 75*, 1–18. https://doi.org/10.1016/j.knosys.2014.07.025
89. Dhiman, G. (2021). SSC: A hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications. *Knowledge-Based Systems, 222*, 106926. https://doi.org/10.1016/j.knosys.2021.106926
90. Abd Elaziz, M., Oliva, D., & Xiong, S. (2017). An improved opposition-based sine cosine algorithm for global optimization. *Expert Systems with Applications, 90*, 484–500. https://doi.org/10.1016/j.eswa.2017.07.043
91. Sree Ranjith, K. S., & Murugan, S. (2017). Memory based Hybrid Dragonfly Algorithm for numerical optimization problems. *Expert Systems with Applications, 83*, 63–78. https://doi.org/10.1016/j.eswa.2017.04.033
92. Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation, 9*, 1–14. https://doi.org/10.1016/j.swevo.2012.09.002
93. Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing, 172*, 371–381. https://doi.org/10.1016/j.neucom.2015.06.083
94. Mirjalili, S., Mirjalili, S. M., & Yang, X.-S. (2013). Binary bat algorithm. *Neural Computing and Applications, 25*, 663–681. https://doi.org/10.1007/s00521-013-1525-5
95. Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002
96. W. Pietruszkiewicz, Dynamical systems and nonlinear Kalman filtering applied in classification, 2008 7th IEEE International Conference on Cybernetic Intelligent Systems, London, UK, 2008, 1–6. https://doi.org/10.1109/UKRICIS.2008.4798948
97. Zhang, X., Xu, Y. T., Yu, C. Y., Heidari, A. A., Li, S. M., Chen, H. L., & Li, C. Y. (2020). Gaussian mutational chaotic fruit fly-built optimization and feature selection. *Expert Systems with Applications, 141*, 112976. https://doi.org/10.1016/j.eswa.2019.112976

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.